

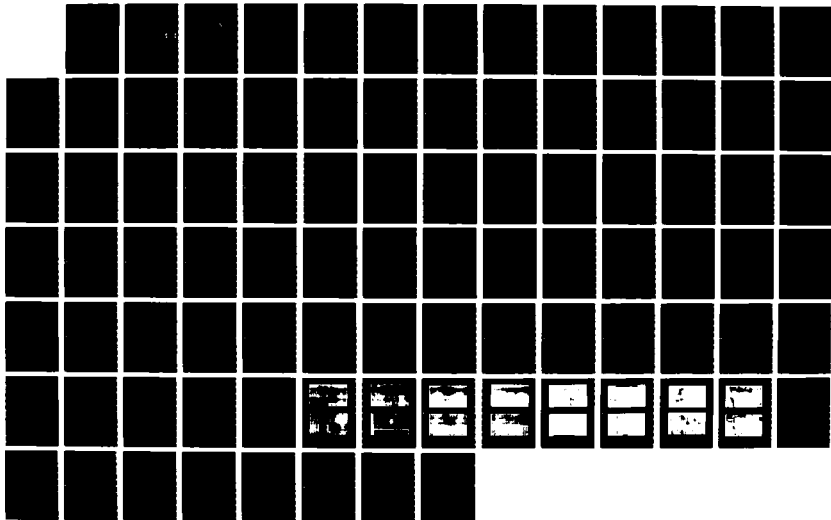
AD-A178 871

A PILOT'S PLANNING AID FOR ROUTE SELECTION AND THREAT  
ANALYSIS IN A TACTI.. (U) AIR FORCE INST OF TECH  
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. J S BRADSHAW  
DEC 86 AFIT/GCS/ENG/86D-11 F/G 6/4

1/1

UNCLASSIFIED

NL





AD-A178 871



DTIC FILE COPY

DTIC  
ELECTE  
APR 10 1987

S  
X

D

A PILOT'S PLANNING AID FOR ROUTE  
SELECTION AND THREAT ANALYSIS  
IN A TACTICAL ENVIRONMENT

THESIS

Jeffrey S. Bradshaw  
Second Lieutenant, USAF

AFIT/GCS/ENG/86D-11

**DISTRIBUTION STATEMENT A**

Approved for public release;  
Distribution Unlimited

DEPARTMENT OF THE AIR FORCE  
AIR UNIVERSITY  
**AIR FORCE INSTITUTE OF TECHNOLOGY**

Wright-Patterson Air Force Base, Ohio

87 4 10 092

AFIT/GCS/ENG/86

DTIC  
ELECTE  
APR 10 1987  
S D D

A PILOT'S PLANNING AID FOR ROUTE  
SELECTION AND THREAT ANALYSIS  
IN A TACTICAL ENVIRONMENT

THESIS

Jeffrey S. Bradshaw  
Second Lieutenant, USAF

AFIT/GCS/ENG/86D-11

Approved for public release; distribution unlimited

A PILOT'S PLANNING AID FOR ROUTE SELECTION AND  
THREAT ANALYSIS IN A TACTICAL ENVIRONMENT

THESIS

Presented to the Faculty of the School of Engineering  
of the Air Force Institute of Technology  
Air University  
In Partial Fulfillment of the  
Requirements for the Degree of  
Master of Science in Computer Systems

Jeffrey S. Bradshaw, B.S.  
Second Lieutenant, USAF

December 1986



Approved for	
DTIC (FAS)	<input checked="" type="checkbox"/>
DTIC (AS)	<input type="checkbox"/>
DTIC (AS)	<input type="checkbox"/>
by	
Date	
Availability Codes	
DTIC	Availability Codes
A-1	

## Preface

The purpose of this study was twofold: to study the use of artificial intelligence programming techniques in a mission planning system, and to upgrade the mission planning system created by Major Robert Bahnij to a prototype operational system for use in a USAFE fighter squadron. Analysis of the accompanying software system in the field will allow fighter squadrons to directly influence the specifications and requirements for future mission planning tools, with the end results being more effective development of software and computers for use in tactical mission planning.

I would like to express my gratitude to Major Steve Cross, my thesis advisor, for his support and assistance with this project. Major Cross was instrumental in getting USAFE to host this software at a 17th Air Force Base, providing a testbed from which a thorough analysis of mission planning techniques can be performed. He also helped nail down the author's follow on assignment from AFIT to the 17th Air Force in Europe, for which the author is greatly indebted. My only regret is that I never got to see the 'Gorilla' crunch a tee shot.

I would also like to show my appreciation to Major Joe Lutz for providing the fighter pilots' insight I needed to understand mission planning. I thank my wife Lisa and daughter Jessica for their love, patience, and understanding as I made my way through the toil of the Air Force Institute of Technology. Lastly, my regards to my softball and golfing buddies who provided some measure of relief during those long, hard 18 months at AFIT.

Jeffrey S. Bradshaw

## Table Of Contents

	Page
Preface . . . . .	ii
List of Figures . . . . .	v
Abstract . . . . .	vi
I. Introduction . . . . .	1
Background . . . . .	1
Problem Definition . . . . .	2
Thesis Objectives and Approach . . . . .	2
Scope . . . . .	4
Assumptions . . . . .	5
Thesis Outline . . . . .	5
II. Background and Related Work . . . . .	7
Overview of the Operational Planning Process . . . . .	7
Unit Level Tactical Mission Planning Systems . . . . .	13
Min Time Tasking Program . . . . .	13
A Fighter Pilot's Intelligent Aide For Tactical Mission Planning . . . . .	15
Mission Support System . . . . .	17
Mission Analysis and Planning System . . . . .	18
Route Planning Aid . . . . .	18
Force Level Mission Planning Systems . . . . .	20
Knowledge Based System . . . . .	20
Tactical Expert Mission Planner . . . . .	21
Force Level Automated Planning System . . . . .	22
Related Planning Systems at AFIT . . . . .	23
Strategy Replanner . . . . .	23
Intelligence Analyst Expert System . . . . .	24
Summary . . . . .	24
III. Conceptual Design . . . . .	26
Desired System . . . . .	26
System Overview . . . . .	29
Mission Card . . . . .	30
The Map Interface . . . . .	31
Intelligence Analysis System . . . . .	32
Stores Management System . . . . .	32
Departure Interface . . . . .	33
Mission Pictogram . . . . .	34
Summary . . . . .	34

IV.	Detailed Design . . . . .	35
	System Specifications . . . . .	35
	Design Methodology . . . . .	36
	Choice of Tool . . . . .	38
	Implementation Details . . . . .	39
	Graphics . . . . .	39
	Planning Functions . . . . .	40
	Planning Database . . . . .	41
	Intelligence Analysis . . . . .	42
	Summary . . . . .	42
V.	System Analysis . . . . .	43
	Design Methodology . . . . .	43
	Human-Computer Interface . . . . .	45
	TMP Critique . . . . .	48
VI.	Conclusions and Recommendations . . . . .	50
	Conclusions . . . . .	50
	Recommendations . . . . .	52
	Operational Implementation . . . . .	52
	Changing the Hardware Environment . . . . .	53
	Map Research . . . . .	53
	Autonomous Planning . . . . .	55
	Contingency Analysis . . . . .	55
	Briefing the TMP . . . . .	56
	Summary . . . . .	57
	Bibliography . . . . .	58
	Additional Sources Not Referenced in the Thesis . . . . .	61
	Appendix A: An Early Version of the TMP . . . . .	63
	Appendix B: TMP User's Manual . . . . .	73
	Vita . . . . .	79



## List Of Figures

	Page
Typical Mission Planning Cycle . . . . .	9
Desired Mission Planning Cycle . . . . .	12
TMP Environment Interfaces . . . . .	28
Modules and Communication Lines of the TMP . . . . .	30
Woffinden's HCI Design Principles . . . . .	46
Incomplete Mission . . . . .	65
Completed Mission . . . . .	65
Departure Window . . . . .	66
Stores Management System . . . . .	66
Stores Menu . . . . .	67
Bomb Menu . . . . .	67
Rack Menu . . . . .	68
New Station Loading . . . . .	68
Color Display . . . . .	69
Message Window . . . . .	69
AGL/Speed Panel . . . . .	70
Color SMS Panel . . . . .	70
Typical Threat Array . . . . .	71
Planning Menu . . . . .	71
Threat Menu . . . . .	72
Display Menu . . . . .	72

## Abstract

A prototype tactical mission planning system is described which provides pilots with automated tools for developing air-to-ground mission plans. This thesis is a continuation of the Fighter Pilot's Intelligent Aide For Tactical Mission Planning.

The Tactical Mission Planner (TMP) contains interfaces to planning and intelligence databases and performs rudimentary threat analysis. Extensive use of graphical displays provide the user with an interactive visual environment for generating mission routes from a starting base to the target and back to the home base. The TMP employs a rapid prototyping design methodology coupled with object oriented programming. This design strategy permits the knowledge engineer to produce an effective planning system without expert knowledge of the planning domain.

# A PILOT'S PLANNING AID FOR ROUTE SELECTION AND THREAT ANALYSIS IN A TACTICAL ENVIRONMENT

## I. Introduction

### Background

A pilot on a tactical mission has two primary goals: mission accomplishment and survival. Often these goals interact in conflicting ways; for instance, a pilot may risk survival if he deduces the only way to destroy his target is to fly through a threat area. However, as the pilot plans the mission, he spends the majority of his time with tasks (fuel consumption, induced drag, turn point calculation, etc.) that degrade situation awareness, preventing direct concentration on the mission goals. These computations can be easily accomplished by a computer, and many attempts have been made at automating the mission planning process. This paper will discuss a mission planning tool that removes the computational burden of sub-level tasks from the pilot, allowing him to focus on the mission as a whole rather than expending time and energy on detached planning details.

This thesis is a continuation of the Fighter Pilot's Intelligent Aide For Tactical Mission Planning created by Major Robert Bahnij (Bahnij, 1985). This work resulted in a prototype mission planning system which showed the feasibility of using basic Artificial Intelligence (AI) programming techniques in a software system that assists a pilot in planning air-to-ground tactical

missions. Demonstrations of Bahnij's thesis sparked an interest in the tactical air force community, and development of a useable system through further research was warranted. By using Bahnij's planning software as a baseline system with specific upgrading and enhancements recommended by operational fighter pilots (Capozzi, 1986), a unit level interactive mission planning system was created in this thesis effort.

### Problem Definition

A working tactical mission planner (TMP) meets an operational requirement for the USAF. The current threat from hostile countries requires a maximum state of readiness for tactical squadrons. The TMP improves readiness by accelerating some of the slow, laborious processes of manual mission planning. The flexibility of the TMP's planning environment permits the exploration of many different flight plans and an overall increase in pilot situation awareness; therefore, the TMP should have a favorable effect on mission success rates by giving pilots more time to assess the high concentration of threats likely to be encountered during a mission.

Creation of a useable planner involves the development of a hardware/software package that fulfills user requirements of being fast, understandable, accurate, maintainable, and easy to use by pilots planning tactical missions.

### Thesis Objectives and Approach

The goal of this research is to design and implement a prototype mission planning system for possible use by an operational squadron in the United States Air Force.

The design of the system is based upon requirements specified by the end users (fighter pilots) (Capozzi, 1986). Provisions for interfacing this system to existing force planning and intelligence processing software are important factors in system design; therefore, a rapid prototyping design approach is employed rather than the traditional 'waterfall' method. Rapid prototyping allows changes of user requirements and software specifications to be easily implemented without major software revisions and excessive documentation overhead. A discussion of both prototyping and waterfall software design methodologies is included in Chapter IV.

This thesis was designed to demonstrate a practical application of artificial intelligence and knowledge based systems in the field of mission planning. Creating a working TMP in the period of time allowed for thesis work shows the flexibility and rapid prototyping capabilities inherent in environments designed for 'smart' software system development. Major Bahnij's thesis work is considered the first prototype and this thesis the second in a series of systems designed for increasing realism, each supporting a wider range of options than were previously available.

The workstation for the mission planner was the Symbolics 3670 Lisp Machine running Zetalisp and the Knowledge Engineering Environment (KEE version 2.2) expert system building tool. The choice of this working environment was based on the availability of useable software and will be discussed later in this thesis.

Evaluation of the mission planner is a continuing project consisting of two phases. A formative analysis was performed as pilots at the Air Force Institute of Technology critiqued the

system; this evaluation was a microcosm of the follow-on project for this thesis. The follow-on project will involve a second and more important evaluation to be conducted when the prototype tactical mission planner is employed in the 496th Tactical Fighter Squadron, an F-16 squadron at Hahn AB, West Germany. The system will be used as a planning tool for F-16 missions and evaluated by the pilots flying these missions. All useful results from these evaluations will be programmed into the planner, creating a constantly evolving system. The final product from this evolution will be a mission planning system from which a set of specifications and requirements can be drawn for future planners. These specifications will be, in effect, written by pilots, since only pilots will evaluate the system. Chapter VI includes a discussion of these specifications.

#### Scope

The overall design of this thesis project was towards a system that is an aid to the pilot, offering options ranging from weapons configurations to suggestions concerning threat avoidance. The end product was an augmented TMP that allows interactive planning of the mission from the starting point to the target. This approach differed drastically from other methods that sought to have the machine plan the entire mission. The intent of this thesis was not to fully automate the planning system, but to always leave the pilot in command of the overall mission plan.

In order to avoid classified material, all information regarding threats and performance models was taken from open

sources. The research effort of this thesis was oriented towards making a valuable contribution to the field of mission planning which would have been hampered by classifying this project.

### Assumptions

It is assumed that the TMP will be deployed to an operational squadron for testing and evaluation over an 18 month period. A formal evaluation of this system will not be included in this thesis; rather, changes suggested through informal evaluations will be annotated here. These evaluations were conducted, using the TMP in its preliminary stages, by pilots attending the Air Force Institute of Technology and other interested parties.

This paper will not discuss in detail the various aspects of artificial intelligence programming; therefore, it is assumed the reader is familiar with this subject. Rich (Rich, 1983), Bahr, Cohen, Feigenbaum (Barr and Feigenbaum, 1981a,b; Cohen and Feigenbaum, 1982), and Winston (Winston 1984) provide excellent sources for artificial intelligence. Also, the reader must have some knowledge of object oriented programming (Pascoe, 1986; Ingalls, 1978; Goldberg, 1984), the Lisp programming language (Winston and Horn, 1984; Wilensky, 1984), and the Symbolics 3600 series lisp machines (Symbolics, 1983). Hearn and Baker (Hearn and Baker, 1986) and Foley and Van Dam (Foley and Van Dam, 1984) contain detailed explanations of the basic graphics techniques used throughout the planning software.

### Thesis Outline

Chapter II will discuss mission planning, including an overview of the operational planning process and software systems

related to tactical planning. Systems covered are the Min Time Tasking Program currently used at Hahn AB, the Fighter Pilot's Intelligent Aide For Tactical Mission Planning, the Mission Support System, the Mission Analysis and Planning System, the Route Planning Aid, the Knowledge Based System, the Tactical Expert Mission Planner, the Force Level Automated Planning System, the Strategy Replanner, and a prototype Intelligence Analyst Expert System developed at the Air Force Institute of Technology. The merits and deficiencies of these systems will be described, along with provisions for overcoming their problems.

Chapter III describes the conceptual design of the mission planner developed during this thesis effort. The desired system is outlined with a discussion of the preferred system architecture. The planner's detailed design is described in Chapter IV. Chapter IV will include a discussion of the design methodology choices available to software designers and the rationale for choosing rapid prototyping as the design approach for this project. Knowledge representation and reasoning abilities of the system will be covered in Chapter IV, along with cases and scenarios that were used in system design. Chapter V describes the first phase of evaluation of the TMP and the incorporation of evaluation results into the final system. Conclusions and recommendations are discussed in Chapter VI, including the plan for implementing the mission planner in an operational squadron.



## II. Background and Related Work

Tactical mission planning is performed in two stages: force level planning and unit level planning. Force level planning is an abstract level of planning in which many flights of aircraft are assigned missions and targets in a large area of operations. Unit level planning is a detailed level of planning in which the abstract force level plans are broken down into separate missions, and flight plans are generated for each mission. This chapter includes an example of how an abstract plan is transformed into a flight plan and a review of computer systems that address various aspects of mission planning.

### Overview of the Operational Planning Process

Tactical squadrons are tasked with missions through an Air Tasking Order or an Air Tasking Message (collectively referred to as an ATO in this thesis). The ATO is generated at a Tactical Air Control Center (TACC) in the Tactical Air Command (TAC) or an Allied Tactical Operations Center (ATOC) in the United States Air Force in Europe (USAFE) under the North Atlantic Treaty Organization (NATO). ATOs are force level plans containing mission assignments encompassing all the aircraft under the direction of the TACC or ATOC.

ATOs do not contain individual aircraft flight plans; they contain mission constraints regarding required times on target (TOTs), desired results, and transit data. The transit data consists of low level transit routes (LLTRs) through friendly territory and transit corridors across the front lines. The

transit data is in compliance with the Airspace Coordination Order (ACO), which restricts aircraft travel in friendly airspace. In a combat environment containing thousands of troops and tanks and hundreds of strike aircraft, adherence to the mission constraints is necessary to optimize the effectiveness of air strikes and to prevent loss of aircraft to friendly fire resulting from misidentification by friendly ground units.

Receipt of the ATO at an air base is the catalyst for unit level tactical mission planning. After being decoded, the ATO is reviewed by the base battle staff and then broken down into individual missions. Each mission is assigned to a flight of pilots and aircraft, and the flight leaders begin planning their missions.

During a state of hostilities, flights may be tasked with missions shortly before the required TOT. The planning cycle for a tactical mission can be as little as two hours, and the majority of this time is spent preflighting the aircraft for the sortie. The flight may have less than an hour to brief and plan the mission. Figure 1 shows a typical planning cycle breakdown from tasking to takeoff.

The following example scenario (Capozzi, 1986) for a battlefield air interdiction (BAI) mission shows the limited amount of time allotted for planning in the overall process. A BAI mission is designed to interrupt the flow of supplies and materiel from the rear echelons to the front lines of the enemy. The target of such a mission will be located between the enemy's forward line of troops (FLOT) and 20 to 30 kilometers into enemy territory. Tasking for this scenario involves a flight of 4 aircraft and is

received from the wing operations center 2.5 hours before the TOT. It will take approximately 30 minutes flight time to reach the target, leaving 2 hours between mission tasking and takeoff.

	<u>Operation Performed</u>	<u>Time Spent</u>	<u>Time Left</u>
	Tasking	-	120 min
M P			
I L	Feasability Check	5 min	115 min
S A	Intel Brief	5 min	110 min
S N	Weather Brief	5 min	105 min
I N	IP Selection	5 min	100 min
O I	Route Selection	15 min	85 min
N N	Mission Brief	5 min	80 min
G			
	Ops - Aircraft Transit	20 min	60 min
	Preflight Check	30 min	30 min
	Engine Start/System Check/Taxi	30 min	0 min
	TAKEOFF	-	-

Figure 1. Typical Mission Planning Cycle

Given this 2 hour planning cycle, a pilot figures that it takes approximately 20 minutes to get from the operations center to the aircraft. This time includes checking personal equipment, putting on an anti-g suit, and traveling to the aircraft. The pilot now has 100 minutes remaining before takeoff.

Preflighting the aircraft involves about 30 minutes of inspecting the aircraft for any problems. Another 30 minutes of time is spent starting the engines, initializing the electrical and navigational systems, and taxiing into position for takeoff. Total aircraft ground operations require 1 hour; time left for in house mission planning: 40 minutes.

The first part of a mission plan will be determination of mission feasibility. The flight leaders (#1 and #3) will check the configuration of the aircraft to be used and the location of the target against a combat mission checklist to determine if the target is within range. If the target is not within the combat radius or the squadron does not have the required ordinance the mission will be cancelled. If the target is within range of aircraft with different configurations then flight lead will ask for a later TOT to allow the aircraft to be reconfigured, assuming that a different configuration will be effective against the given target. Flight lead may have to ask for a later TOT if the tasking is too late for the aircraft to reach the target at the given TOT. Resolving these questions to determine if the mission can be performed will take roughly 5 minutes.

There are two briefings that a pilot will encounter during the remaining 35 minutes. The first is an intelligence (Intel) briefing, in which the pilot is briefed on the enemy's ground and air orders of battle, known threat locations, and recent Intel updates. During this briefing, the pilot will discuss LLTRs, transit corridors, and the transit level to find the safest route across both friendly and hostile FLOTs. If the pilot is familiar with the area of operations and the majority of threats to be encountered, such a briefing should take approximately 5 minutes. After the Intel brief, the pilot will receive a weather briefing taking another 5 minutes. He now has 25 minutes left to plan the mission.

There are many different tasks to be performed during this 25 minutes, depending on the element of the flight to which the

pilot is assigned. The flight leaders select the initial point (IP - the point at which the pilot will update all his navigation and weapon systems and begin his attack upon the target) from detailed aeronautical charts. Leaders will plan attack tactics and egress from the target and give this information to their wingmen. It will take the flight leaders approximately 5 minutes to select the IP and attack parameters, which must be completed before the wingmen begin selecting turnpoints and planning fly-able routes.

The wingmen (#2 and #4) will plot the LLTR on a large scale aeronautical chart, along with the target and IP recieved from the flight lead. They will take coordinates and elevations of turnpoints and landmarks from the aeronautical chart in order to program the inertial navigation system (INS) of the aircraft. These coordinates must be accurate, since the pilot uses them not only to navigate but also to attack the target. Also, if the pilot becomes lost (avoiding a missile or other threat) flying 100 feet off the ground at 400+ miles per hour, the INS will allow him to quickly get back on track. Since the INS is of such great importance, the wingmen will choose offset aim points (OAPs) and visual reference points (VRPs) that provide backup landmarks and headings for the pilot to return to his planned route.

After choosing and plotting turnpoints, OAPs, and VRPs, the wingmen will compute the distances between turnpoints along with the fuel used for each leg of the route. The time of arrival at each turn point will be determined, and all of this information (times, fuel, distances, and coordinates) will be placed on a

mission planning card. After the mission cards are complete, the wingmen will reproduce 4 strip maps and 4 cards to complete the mission plan. This period of detailed planning takes around 15 minutes; including the 5 minutes for choosing the IP, the pilots are left with 5 minutes for a briefing to enhance situation awareness.

<u>Operation Performed</u>	<u>Time Spent</u>	<u>Time Left</u>
Tasking	-	120 min
Computer Aided		
Mission Planning	10 min	110 min
Mission Brief	30 min	80 min
Ops - Aircraft Transit	20 min	60 min
Preflight Check	30 min	30 min
Engine Start/System		
Check/Taxi	30 min	0 min
TAKEOFF	-	-

Figure 2. Desired Mission Planning Cycle

Although this is a scenario, it is not unrealistic. Many missions are planned on short notice with little time spent discussing how the mission will actually be flown. Current automated planning systems can accomplish route selection in about 5 minutes, giving pilots an extra 10 minutes to brief the mission. If the whole of the mission planning phase of the planning cycle can be accelerated with computer assistance, pilots would expend less time with disconnected planning subtasks and more time discussing the mission. Using a computer aided planning system to automate the feasibility check, display

current weather and intelligence information, and assist with selection of both the IP and the mission route results in a planning cycle that gives pilots more time for briefing and increasing situation awareness than was previously available, as shown in Figure 2.

#### Unit Level Tactical Mission Planning Systems

The standard method of mission planning involves using grease pencils to draw mission legs between turn points on a mission map. Drag indices and gross weights are acquired from mission planning handbooks. Headings, distances, and fuel consumption are measured using straight edged planning templates and hand held calculators; this information is recorded on an Air Force Form 70 (Mission Data Card) to be loaded by hand into the aircraft's avionics systems. Threats are placed on mission maps by drawing circles with the grease pencil. Since this method is somewhat primitive, the USAF has begun placing microcomputers in squadrons as mission planning aides.

Min Time Tasking Program. The computers placed in operational squadrons are 8-bit microcomputers running BASIC programs developed by each squadron for mission planning. One of these programs is the Min Time Tasking Program developed by Captain Alex Rupp for the 313th Tactical Fighter Squadron, Hahn AFB, West Germany (Rupp, 1986). This program was designed to assist pilots in quickly planning F-16 missions.

The Min Time Tasking Program allows pilots to plan missions from turnpoint to turnpoint with entry of these points through latitude/longitude coordinates (lat/longs), universal terrain

masking coordinates (UTMs), or from preplanned points stored in memory. The computer calculates bearing, range, time, turn radii and fuel consumption for each leg between turnpoints. The system also keeps running real and bingo fuel; the real fuel table shows the amount of fuel the aircraft should have at any point along the mission, and bingo fuel shows the minimum amount required at that point to complete the mission safely. All planning information is stored on a mission card, and the system can print 1 to 8 copies of the card to be carried to the aircraft.

The Min Time Tasking Program works well but has a number of limitations. The system ignores aircraft configuration, requiring the pilot to input the fuel flow for the entire mission. Only one fuel flow is entered; consequently, the running real and bingo fuel tables are very rough estimates. The planner allows only one change in speed from the IP to the target, and automatically assumes that the aircraft will resume its original speed after carrying out its attack. The planner requires use of an aeronautical chart, and the pilot must pick his turnpoints off of this map before entering them into the computer. Although this seems like a trivial task, it is this portion of mission planning that consumes the greatest amount of time due to the required precision of the waypoints for the INS (Capozzi, 1986). Therefore, the Min Time Tasking Program is limited in effectiveness since it does not address the problems associated with manual turnpoint selection. The associated shortcomings of this system have been addressed in the prototype planner developed at AFIT and in systems developed through the Tactical Air Forces Interoperability Group (TAFIG).



The Fighter Pilot's Intelligent Aide for Tactical Mission Planning. Major Robert Bahnij's mission planning tool (hereafter referred to as Bahnij's Planner) is a menu driven planning system with a graphical interface showing a map of the area of operations for fighter missions (Bahnij, 1985). The system resides on a Symbolics 3670 Lisp Machine with a tandem color monitor and a mouse, running the Knowledge Engineering Environment (KEE) and Zetalisp. Using the mouse, the pilot draws his mission route on the map (displayed on the black and white monitor) and the computer calculates the amount of fuel used between turn points. The system shows surface-to-air missile (SAM) sites, the target area, terrain contours, and mission data (fuel used and distance traveled).

A sample session using Bahnij's system begins with the computer terminal displaying a contour map of the area of operations for the mission. The color monitor to the side of the terminal displays mission information and leg information. The user places the mouse over the contour map and presses the left button (clicks the mouse) at which point a menu appears. The user chooses the 'Select Start-Point' menu option and clicks the mouse at the point where he wishes to begin his mission planning. The machine calculates the amount of fuel used, distance, and flight time from take-off to this starting point and displays this information on the color monitor. The user now clicks left on the mouse, chooses 'Build A Leg To A Turn Point', and clicks the mouse at the place he wishes for his first waypoint. The user again chooses 'Build A Leg...' and repeats this process until he has finished planning his route. If he wishes, he may place a SAM on the map by choosing an option from clicking the middle

mouse button. The planner supports only a generic SAM object, which has no special connotations other than being a threat to mission survival. The color monitor may be used for displaying the line of sight of the SAM or the field of view of the pilot at any point on the map.

The artificial intelligence programming techniques used in Bahnij's Planner consist of object oriented programming implemented through the use of a frame structured knowledge base. Information for each of the units in the system (the pilot's plane, mission legs, SAMs) is stored in objects called frames. KEE uses frames to represent knowledge in a hierarchy of levels in which higher level frames hold information that is more generic than the detailed information found in lower level frames. This use of frames saves space and allows knowledge to be stored at abstract levels in the data base. Fikes and Kehler (Fikes and Kehler, 1985) offer a good description of frames and their uses.

Bahnij's Planner is a useful feasibility model for mission planning systems supporting a full range of planning functions, including a planning map for turnpoint selection. However, this system is not useful as an operational system and was not intended for such use (Bahnij, 1985: I-4); rather, it serves as a prototype for more advanced systems.

There are a number of problems associated with Bahnij's Planner that need to be addressed in future systems. While this system employs a mission map for route selection, the map is essentially unuseable in an operational environment. The map displays terrain contours only, having no features from which to choose turnpoints. The system has limited capability for dealing

with threats through a generic SAM object, which is unrealistic for an operational planner. Finally, Bahnij's Planner has a faulty performance model that does not calculate correct mission parameters for its target aircraft, the F-16.

Mission Support System. TAFIG is developing a mission planning system for tactical F-16 missions (Simpson, 1986). The Mission Support System (MSS) will consist of two separate computer systems: the Mission Planning System (MPS) and the Data Transfer Cartridge Loader/Reader (DTCLR). The MSS will employ the MPS to produce a mission plan; after the plan is generated, the DTCLR will load aircraft specific avionics information and the mission plan onto a cartridge tape which will be read by a tape reader onboard the aircraft. This electronic transfer of the mission plans from the MPS to the aircraft computers will reduce the time required for ground operations and will eliminate the possibility of human error in transferring mission data between systems.

The MPS software will reside on a Cromemco CS220 computer with a 68020 microprocessor, a 150 megabyte hard disk, and a digitizer board. The MPS will be an integrated package of 3 software modules. The flight planning module will assist with route planning; waypoints will be selected via the digitizer board overlaid with standard aeronautical maps. The penetration analysis module will allow input and display of threats with terrain masking of these threats at 3 altitudes. A dynamic programming routine will search to the target and back at a given altitude while minimizing the route's lethality, and the

weaponeering module will calculate the conditions and parameters for various weapon delivery modes.

The MSS system is scheduled to be delivered to TAFIG in December of 1987. The system will be deployed to operational units after acceptance by TAFIG. Since the system is not yet in use, no evaluations can be made of an actual working MSS. However, pilots criticize the use of digitizer boards because the board is a large piece of hardware that is difficult to use and takes up too much space in the squadron operations center (Lutz, 1986). TAFIG recognizes this problem and plans on implementing a digitized map using optical disk technology in the MSS in a future release.

Mission Analysis and Planning System. Optical disks are in use in Fairchild Communications and Electronics Company's Mission Analysis and Planning System (MAPS) (Aviation Week and Space Technology, 1986:97). MAPS performs the standard planning functions (turnpoint selection; leg generation; fuel, distance, time, and heading computations) with aeronautical charts displayed by the computer and can print a hardcopy of the generated route. The MAPS system may be interactive with the pilot/planner, or the computer can autonomously find penetration routes to the target given threat and aircraft configurations. MAPS is under current development and future versions will be oriented towards placing MAPS in the cockpit for inflight planning use.

Route Planning Aid. The Route Planning Aid (RPA) (Rockmore and others, 1983) is a system that allows both interactive and automatic route generation. RPA contains a knowledge based system with route optimization, terrain masking, threat analysis,

and evaluation algorithms that permit generation of routes and critiques of those routes.

RPA provides the user with the capability of controlling route generation through the system executive. If the user generates the mission route, the system can critique it for survivability; if the system generates the route, it will also present an explanation of why the turnpoints were chosen and which legs are under threat. The system permits the user to examine the route in detail, or to choose alternate routes for contingency analysis.

RPA's ability to evaluate and critique routes is useful for automatic route generation. When given a route derived by a computer, a pilot will be skeptical of the validity and safety of the route and will proceed to check the route for problems and to familiarize himself with the mission. The computer is alleviating the pilots' doubts and enhancing his situation awareness by providing him with a critique of the route which serves as pre-flight route briefing.

The RPA system resides on a VAX minicomputer running under the VMS operating system. The knowledge based portion of the system is written in Interlisp; the optimization algorithms are implemented in Fortran. RPA requires both jobs to run concurrently on the computer and communication between the jobs is performed by reading and writing commands to and from a file shared by both processes (Rockmore and others, 1983:7). While RPA provides a tool for use in mission planning research, its effectiveness as an operational system is hampered by its implementation on the VAX, a system too large to be placed in every fighter squadron.

### Force Level Mission Planning Systems

Although the emphasis of this thesis is towards squadron level planning systems, it is important to note the work done in force level planners. These planners are designed for producing ATOs for coordinated use of the air power available in a theater of operations. Since the ATO is the impetus for the unit level planners, provisions need to be made for direct transfer of the ATO from the force level planners to the unit level planners. Force level planners must perform real time threat analysis in generating an ATO, and an interface to both threat analysis and ATO data is required for the TMP to be an effective planning system.

Knowledge Based System. Knowledge Based System (KNOBS) is an interactive experimental force level mission planner developed at the MITRE Corporation (Engelman et al., 1983:450). KNOBS has been used for planning missions for an offensive counter air battle incorporating many sorties of aircraft, and it has proved useful in planning Naval task force deployment, generating activity plans for space shuttle crews, and planning space shuttle fuel loading tasks.

The end product of planning with KNOBS is an ATO. Tactical missions are planned with KNOBS by matching staging bases and aircraft with required missions. KNOBS implements threat analysis and contingency planning can be performed by eliminating friendly bases from the available planning assets. Automatic replanning occurs during contingency analysis.

KNOBS is a menu driven system that uses an English language parser to understand commands from the user and displays current

mission information in text. The knowledge base is implemented through frames, and constraints upon frame slots permit consistency checking and problem detection in the mission plan. Research and further development of KNOBS has produced an operational successor, the Tactical Expert Mission Planner (TEMPLAR).

The Tactical Expert Mission Planner. The TEMPLAR system is in advanced development and is intended for use in the 9th Air Force at Shaw AFB, South Carolina (Shapiro and others, 1984). TEMPLAR uses the basic architecture of KNOBS with modifications for improved performance. The TEMPLAR workstation is the Symbolics 3600 series Lisp machine with both color and monochrome monitors, a mouse, a 300+ megabyte disk, and a printer. The system is implemented in Common Lisp.

TEMPLAR works with templates hierarchically arranged: the top level template represents the ATO, intermediate level templates represent strike packages, and low level templates represent missions (Shapiro and others, 1984: 1-8,9). The system fills in these templates using a rule based expert system employing backward and forward chaining, inferencing through inheritance, and constraint checking. The ATO is completed when all lower level templates are fully instantiated.

The goals of TEMPLAR are (Shapiro and others, 1984:1-4): to reduce the planning time from target identification to the production of the ATO, to reduce planning errors, to improve plan quality, and to reduce manual processing. The system analyzes aircraft constraints, mission types, targets, and intelligence data to produce an ATO along with some analysis of the ATO.

TEMPLAR monitors the user's planning decisions and assists in entering targets and weaponeering. Future directions for TEMPLAR are automated evaluation of the ATO quality, resource optimization, and possible English language interaction with the system.

Force Level Automated Planning System. The Force Level Automated Planning System (FLAPS) (Rainbolt, 1985) is a program by System Control Technology, Inc., for generating an ATO. FLAPS was developed for USAFE, and although the system was not intended for operational use it has been employed in support of mission planning in USAFE.

FLAPS works in six basic steps: 1) update the ACO, force status, target, and Intel databases; 2) match targets to bases; 3) compute optimal routes to and from targets, 4) allocate weapons to targets, 5) open flight corridors using support aircraft for threat suppression, and 6) print the ATO (Rainbolt, 1985: III-2). Steps 3, 4, and 5 can be repeated until an acceptable ATO is found. The system uses dynamic programming techniques with performance and threat models to find routes and allocate threat suppression.

Systems such as TEMPLAR and FLAPS have the capability for generating unit level plans; however, such plans are not always flyable by aircraft. The routes generated by these systems tend to have many turnpoints and sharp turns (Rainbolt, 1985:VII-3). There is also no guarantee that the turnpoints will be readily identifiable by the pilot from the air. A system that allows the user to adjust the routes to fit his needs would solve this problem; such a capability exists in the TMP created for this thesis.



### Related Planning Systems at AFIT

Two mission planning systems at the Air Force Institute of Technology show applications of artificial intelligence programming to the mission planning field. The Strategy Replanner (Smith, 1986) and the Intelligence Analyst Expert System (IAES) (Foley, 1986; Phillips, 1986; Bradshaw, 1986) address the problems associated with considering threats in planning tactical missions.

Strategy Replanner. Captain Mike Smith developed a planning system that uniquely represents threats and performs a search through the threats for a legitimate route from a given point to the target (Smith, 1986). Smith's system represents a threat as a square, and each corner of the square is a point in the search space. Points that lie within other threats are considered covered and are excluded from the search space. The planner is capable of performing a number of searches (Best First, A\*, Beam\*, etc.) through these points in an attempt to find a threat free route to the target. If no route can be found, the planner will 'relax' threats by including points covered by threats that can be defended against using electronic countermeasures. The Strategy Replanner will continue to relax threats and search the threat space until a route is found to the target.

The benefit from representing threats as a set of points is the size of the search space is much smaller than a search space resulting from the traditional 'grid' composed of terrain divided into sectors with a threat value assigned to each sector. However, the Replanner does not consider terrain nor terrain masking

effects. This shortcoming is addressed in Chapter VI along with integration of the Strategy Replanner and the TMP.

Intelligence Analyst Expert System. The IAES was developed as a rule based system for threat analysis for air-to-ground missions (Foley, 1986; Phillips, 1986; Bradshaw, 1986). The system was designed to predict and display threat movements over a short period of time. Using simple rules and models of threats, the IAES repositioned known threats to probable locations in which the threats enhanced their ability to cover their areas of responsibility. For example, reasoning that an SA-6 SAM battery would attempt to cover a mobile asset (such as an enemy command post) and knowing that missiles prefer 'high ground' from which there is an unobstructed field of view for clear target acquisition, the IAES would reposition the battery to the highest traversable terrain from which the SAM could provide protection of the asset and the field of view was clear towards the FLOT. While the IAES could not provide a guaranteed location of mobile threats, it did provide research into terrain analysis and reasoning with uncertainty. The IAES was a modification of the mission planner created by Major Robert Bahnij, showing the flexibility of Bahnij's Planner and the extensibility of Lisp based prototypes.

#### Summary

By using the main ideas or capabilities of the systems discussed in this chapter, the TMP can provide pilots with information and assistance as missions are planned. Threat analysis is of substantial importance, and the TMP provides an interface to a

system such as FLAPS or TEMPLAR for real time threat analysis data. Use of explanation and route analysis capabilities similar to those found in RPA are desirable and should be included in future versions of the TMP. These and other capabilities (i.e., autonomous planning) are considered in Chapter VI as future enhancements.

Tactical mission planning is currently a labor intensive process in which the planner is inundated with tasks that prevent him from allocating an appropriate amount of time to briefings that serve to increase his situation awareness. Many tools have been created that address this problem, and the goal of providing pilots powerful planning environments at the squadron level can now be reached. If these environments and tools are sufficiently flexible to accomodate current and future needs of pilots and planners of tactical missions, the planning cycle can be used by pilots to thoroughly prepare for the mission at hand instead of spending time with seemingly unrelated tasks that tend to impair their situation awareness.

### III. Conceptual Design

The mission planning system described in this thesis was conceived for use by pilots as a planning system for low level tactical missions. Its requirements were accuracy (since expensive aircraft and human life may rely on the output), speed (the TMP must be faster than humans in planning a full mission), user friendliness (or pilots will not use it), and understandability. The TMP uses these four requirements together with six minimum planning capabilities as an initial abstract specification for the conceptual system design.

#### Desired System

There are six minimum capabilities of an operational mission planning system. These are (Foxwell, 1986):

- A. Store a file or files of turn points for planning use.
- B. Save/recall completed mission plans/routes.
- C. Convert Coordinates from UTM's to Latitude/Longitude.
- D. Calculating VIP/VRP data.
- E. Printing (output) a useful mission card.
- F. Work faster than a four man flight planning by hand.

These capabilities are not difficult to program into a computer; only the speed requirement poses any hard constraints. Storing files of points and completed missions for later recall are minor database operations that require simple algorithms for implementation. Converting coordinates from the Army Universal Terrain Mapping (UTM) system to latitudes and longitudes requires a more involved algorithm but is straight forward. Calculating visual identification points (VIPs) and visual reference points (VRPs)

is done using the same method for choosing a waypoint along the mission route. Printing a mission data card can be done using a screen dump of the displayed flight card, or a serial output stream can also accomplish this task. Since these tasks present no foreseeable problems, further capabilities can be added to the planner such as the ability to assimilate information contained in the tasking orders including the target, TOT, LLTRs, and ordinance. Both standard and nonstandard aircraft configurations, selectable by the system user, are needed along with maps and aircraft performance data. A planning environment familiar to the end user, a pilot, would exist by creating a system with the above capabilities.

There will be a number of interfaces required for a full scale planning system. These include an ATO input capability in order for the machine to accept tasking from the Allied Tactical Operations Center (ATOC). By using the ordinance required in the ATO and the distance to the target, the system will be able to determine if the squadron is equipped to perform the mission. If the mission is acceptable, the TMP should be able to select the correct maps for use by the pilot using the ATO's target and LLTR information.

A second required interface will be to the intelligence database. Information concerning threats, enemy and friendly front lines, and areas of threat suppression will be presented in a format tailored to the user's needs. Threats may be displayed by individual units and their maximum range of lethality, by probability of kill contours generated by intelligence processing software, or a mixture of both. The pilot may wish to suppress

all threats except those that directly affect his mission. This choice of threat displays will prevent the screen from becoming saturated with unnecessary information and distracting the pilot from his mission.

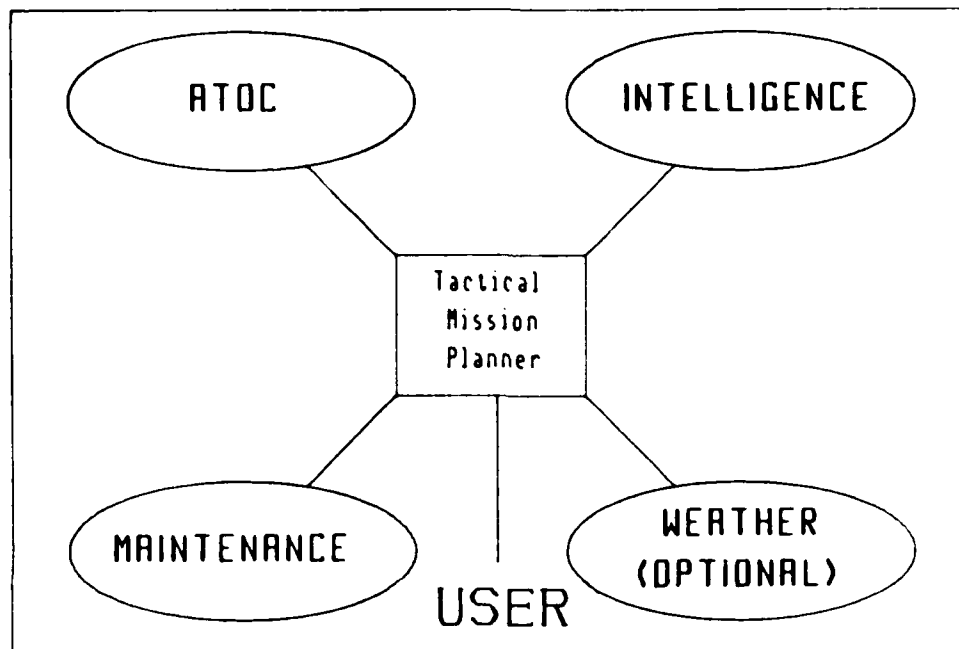


Figure 3. TMP Environment Interfaces

A third interface will be to a maintenance database that shows the condition and availability of aircraft in the squadron. This interface will tell a user if the aircraft he is to fly is mission capable, and if the ordinance loaded on the airplane is compatible with the mission goals. Such an interface will be used to avoid the problems of flying to destroy a 'hard' target (i.e., a bridge) with 'soft' weapons (i.e., anti-personnel bombs). Figure 3 shows the environment in which the planner will exist, including the three desired interfaces and information received through the interfaces. An optional fourth interface to weather

data is included in the figure; although meteorological data is a valuable asset to planners its incorporation into the TMP is beyond the scope of this thesis.

The TMP is designed as an interactive system, not as an autonomous system. The user interface provides the pilot access to the major components of the TMP and gives him control of the planning process. In systems that automatically generate the mission route, pilots must spend time familiarizing themselves with the turnpoints and route parameters. This 'self briefing' must take place after the computer has produced the mission plan, and the pilot is left out of the plan generation process. When the system permits interaction with the planner, the pilot increases his situation awareness during generation of the route (Bahnij, 1985:IV-13). The pilot 'briefs himself' as the route is chosen; he becomes familiar with the turnpoints, terrain, and threats and can spend time thinking about flying the route while the computer does the mathematical calculations associated with leg generation. By using the user interface as the main executive controlled by the pilot, the planning system enhances the pilots' knowledge of the mission.

#### System Overview

The interfaces and capabilities mentioned in the previous section show the desired communication of the TMP with the outside world. To the system user, the TMP is a 'black box' which manipulates data and commands into a mission plan. In order to implement the planning capabilities in this 'black box', the TMP

must provide the same tools that a manual mission planner possesses: a map, a mission card, and planning formulas for computations. The information from the interfaces and the planning tools can be incorporated into the TMP in six separate areas: a mission card processing system, a map/route selection interface, an intelligence processing and analysis system, an aircraft configuration management system, a departure selection system, and a mission pictogram display. Figure 4 shows the main modules (tools) in the TMP, with the arrows showing the direction of communication between the tools.

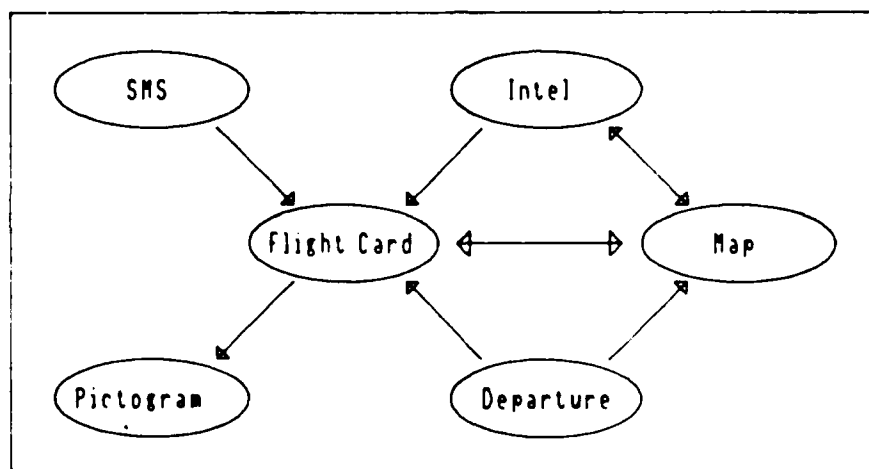


Figure 4. Modules and Communication Lines of the TMP.

#### Mission Card

The mission card serves as the primary interface for most of the planning functions of the TMP. The card displays the current state of the mission and can be thought of as a window into the mission database. The mission card has both input and output capabilities to the other areas of the planner, permitting parts of the route to be input from various sources.



Inputs to the card may be through the ATO (target and LLTR), the maintenance interface/SMS, the map interface, or directly from the pilot. The purpose of these various inputs are to allow the pilot flexibility in building his route. Given a mission with a specified LLTR, the pilot should be able to choose his departure from home base, select points to the first waypoint of the LLTR by simply entering names of stored turnpoints through the mission card, complete the route to the IP by selecting landmarks from the map display as turnpoints, and then use some combination of map and card waypoint selection to complete the mission.

After the mission is completely planned, the pilot requires the option of changing waypoints and other parts of the mission. In the case where a change is made, the TMP should update all related aspects of the mission. Changes may also be made by changing points through the flight card, or by moving parts of the route through the map interface.

#### The Map Interface

The map interface will display digitized aeronautical charts from which turnpoints and mission legs can be chosen. The current position of the front lines and the threat array will be shown on the map so that the pilot will be fully aware of obstacles to successful plans. The map will provide basic planning functions through a graphical interface; these functions are: 1) building mission legs; 2) picking latitudes, longitudes, and elevations of points on the map; and 3) modifying the route. The map will also provide threat displays produced through an interface.

to the Intelligence Analysis System which will perform rudimentary threat analysis.

#### Intelligence Analysis System

The threat database will be evaluated by the intelligence analysis system (IAS). The IAS will provide links to real time intelligence databases, and will perform simple analysis of these databases. Threat analysis and display functions will permit placing surface-to-air missile (SAM) and anti-aircraft artillery (AAA) batteries, moving selected threat units, displaying threats as individual threat rings or probability of kill contours, and analyzing threats in order to defend against them when mission legs fall within their effective range. The IAS will be tightly coupled to the map interface for its output.

#### Stores Management System

The method in which an aircraft is configured (loaded) with ordnance determines the performance of the aircraft during the course of a mission. A standard configuration load (SCL) is an external stores loading which has been evaluated through flight test. SCLs provide the mission planner with drag indices, gross weights, performance limitations, standard fuses and ordnance delivery modes. From the SCL's drag index and gross weight, a planner can determine his maximum g loading and fuel flow. Fuse information is used for onboard programming of the aircraft's weapon systems. The delivery modes for the weapons will specify the attack parameters required for releasing the ordnance and avoiding fragmentation and blast effects from the bomb explosions. This information provides initialization for performing

accurate planning calculations of both route data and target area tactics.

The Air Tasking Order contains information concerning the type of weapons needed to destroy the given target. This information may be a specific SCL (for example, Mk 82 Snakeye bombs) or general ordinance (BA or Best Available ordinance). In either case, aircraft on the flight line may not be configured to the ATO specification. These situations are taken into consideration in a Stores Management System (SMS). In a situation in which the ATO contains a specific SCL, the SMS will search its database for this configuration and initialize the planner accordingly. If the squadron does not have the correct weapons or the aircraft on the flight line are loaded differently, the pilot may have to change the configuration, entering his new SCL through the SMS. If the SMS is correctly implemented through the maintenance interface, the TMP will be able to automatically configure the aircraft performance model to reflect the current loading of aircraft on the flight line.

#### Departure Interface

The departure interface allows the pilot to choose his first few waypoints through an enlarged view of the area surrounding the home base. This interface shows the pilot the fastest way of getting out of local air traffic and heading towards the target. In a TMP that has a weather database, winds, gross weight, and drag index can be used to determine the departure from the airfield. However, since weather will not currently be implemented, a departure interface is provided to select the initial route legs.

### Mission Pictogram

The mission pictogram is 'programmed' by the TMP as the pilot completes his mission plan. After planning is complete, the pilot chooses to display his overall route on the pictogram. The resulting display includes a drawing of the route with threats associated leg data (heading, description, leg distance). Also, vectors to the nearest friendly base for each leg are displayed, giving true bearing, distance, and associated radio frequencies. These vectors are provided in case an emergency situation occurs along the route. The primary weapons delivery mode is displayed with a pop-up, apex, and release drawing as a reminder to the pilot of his preplanned target area tactics.

The pictogram serves as a quick reference for the pilot, so any useful mission information is included on the display. The majority of pictogram information is computed through the planning interface to the mission card or by direct input from the pilot. When the pilot has chosen his mission plan and reconstructed his route with the pictogram, mission planning is completed.

### Summary

This chapter has defined the basic abstract requirements of tactical mission planners. It has also specified a toolset for implementing those requirements while giving the pilots flexibility in planning their missions. Chapter IV will discuss the design methodology of the tool, the choice of the hardware environment, and the detailed design of the system.

#### IV. Detailed Design

The mission planning software created in this thesis work is a straight forward implementation of object oriented programming, graphics programming, and simple constraint checking. The system hardware provides a powerful environment that supports very fast planning with large terrain databases. This chapter will discuss the creation of the system specifications, the design methodology, the choice of the hardware environment, and implementation details of the TMP.

##### System Specifications

The system specifications for the tactical mission planner were generated through a series of interviews with Major Rocky Capozzi, an F-16 pilot and member of the headquarters staff of the 17th Air Force, USAFE (Capozzi, 1986). These interviews consisted of analyzing current mission planning systems in USAFE and at AFIT. Rough specifications of the planning system resulted from this analysis. After considering the available hardware and tools at AFIT, the specifications turned into the conceptual design discussed in the previous chapter. Since the specifications and design were drawn from a single pilot's opinions, it was determined that the resulting system should be flexible in its implementation in order to accept modifications suggested by other pilots. The planner would be required to interface with many different databases and force planning systems, making further demands on its flexibility. This flexibility can be implemented by using a rapid prototyping design methodology.

## Design Methodology

There are a number of different methods for creating software for use over a long period of time; two of the most widely used approaches are the life-cycle or 'waterfall' (named for the similarity of the design to water flowing over steps) model for software engineering and evolutionary programming/rapid prototyping. Figure 5 shows the steps in software development for both models. The two methodologies share steps in the latter stages of software development and maintenance yet differ in the formalization of system requirements and specifications.

In the waterfall model, a set of requirements that loosely describe a given problem are defined. After all the requirements are set forth, specifications describing the problem in depth are created. Once these requirements and specifications are determined to completely cover all the aspects of the problem, a system design is proposed. The design is placed into code, tested, and if the code is correct, the system is released for use.

The waterfall model works well for applications that are well defined. For instance, a numerical analysis program is a stable problem that is well suited to the life cycle model. A payroll program is another example of an application in which the requirements and specifications can be easily drawn. However, problems that are not easily specified pose a dilemma for the life cycle model of system design. When the specifications are incomplete, the resulting software design cannot be robust, and the system has imperfections in it that tend to pose difficult problems in the maintenance phase of the software life-cycle.

If the difficulty lies not in incomplete specifications but in a change or misinterpretation of requirements after coding has begun, the waterfall model poses a resistance to the required changes, and code modifications can be extensive.

In a situation in which the problem defies clear definition or changes to the requirements are likely to occur, a rapid prototyping approach can be used. In rapid prototyping, a quickly designed system that has the potential for solving the problem is presented to the end user. If the prototype meets the needs of the user, a series of specifications is taken from the prototype and full scale design and implementation are performed. If the user requires a change, the prototype is manipulated to perform the new requested function. This process of rapidly modifying a prototype to meet the needs of the user allows the user to actively participate in the requirements and specifications definition while getting a feel of what the final system will look like. Rapid prototyping is amenable to change and provides an alternative approach to the waterfall model of design.

Rapid prototyping was chosen as the design methodology for the TMP for three reasons: the system developer was unfamiliar with the problem domain, resulting in changes due to lack of understanding of the task at hand; new requirements were defined as the system began development; and the waterfall method prevented flexibility in dealing with the requirements of a number of different users (fighter pilots at AFIT). The environment and implementation language also supported rapid prototyping as opposed to strict adherence to specifications.

### Choice of Tool

Pilots planning missions require a map and flight card to prepare a mission plan. Using a computer to display a map that provides a large terrain area for planning requires supporting a large array for the terrain data. The TMP must be able to support this array and allow pilot interaction through a graphics based interface, since pilots are visually oriented. The Lisp language is an interactive language, and the Symbolics 3600 series lisp machine is capable of supporting and processing large arrays and bitmapped graphics. By analyzing planning problems, it was determined that object oriented programming closely resembles the method in which pilots think about planning. Pilots consider threats as threat objects, as they do turnpoints and mission legs. The Symbolics 3600 supports Flavors, an object oriented programming package, and the Knowledge Engineering Environment which uses hierarchically structured frames for objects. KEE provides constraint checking; for example, constraints prevent the pilot from running out of fuel as he plans his mission. KEE contains a rule interpreter for the possibility of building an expert system on top of the planner. Since the programming environment on the Symbolics 3600 met all requirements for the TMP, it was chosen as the hardware support for the system.

Other environments available during the early development of the TMP were microcomputers and minicomputers. The microcomputers were determined to have insufficient memory in the available configurations to effectively process the map data included in the TMP. The minicomputers available were DEC VAX 11/780 machines running UNIX and VMS operating systems. These machines are



used for programming support at AFIT and insufficient space could be allotted for the TMP. No other environments available at AFIT were as well integrated as the Symbolics 3600, and few possessed the required processing capabilities.

#### Implementation Details

As mentioned at the beginning of this chapter, the TMP's implementation details involve simple processes and algorithms for calculating and displaying information for the user. This section will discuss these functions at an abstract level of detail, and in those cases where the algorithms are involved, more information will be given.

Graphics. The majority of graphics displays are accomplished by creating a window on the screen, performing a bit-block transfer (bitblt) of a stored bitmap onto the window, and defining areas of the window to be active when pointed at by the mouse. KEE provides mouse sensitive panels, and the Flavors package allows the user to define window object types (referred to as the 'flavor' of a window) and messages ('methods') that act upon the flavor. Both packages also provide for menus that can be accessed through the mouse.

The actual implementation of line drawing functions is through the non-standard Flavors package. This package permits a line drawing message to be sent to a window. Such a message will consist of the type of algorithm (draw-line, draw-circle, etc.) and parameters for the message. The parameters are the physical coordinates on the window at which the lines, circles, or points will be drawn. Flavors performs a mapping of the position of the

window into the output device, allowing the user to not worry about translations from virtual to physical devices.

Turnpoints are displayed by a draw-circle message; legs with a draw-line message. Threats are displayed by bit-blt'ing the threat bitmap onto the window and sending a draw-circle message to the window with the effective range of the threat as the circle radius. The intelligence interface provides the data for threat and FLOT locations. The FLOT and probability of kill contours are implemented by the draw-cubic-spline message provided by Flavors. The points used in the cubic spline algorithm are saved in arrays of (x,y) coordinates for display and use with the planning functions.

Planning Functions. The planning functions use domain specific information to provide them with information necessary for performing route parameter calculations. Where possible, this information is placed in global variables that can be quickly altered by the system designer. While this may appear to be poor programming practice, it is essential for the flexibility of the planner. The Lisp language implementation on the Symbolics 3600 through Zetalisp encourages the use of global variables for both space and time considerations, and it is much easier to set a global variable than to redefine a planning function due to changes in the planning domain.

Turnpoints and mission legs are defined as flavors, and a list is used to keep track of instantiations of these objects. Turnpoints are selected through the flightcard by accessing mouse-sensitive areas of the card. Points are selected through the mission map by selecting a function from a menu and then clicking

the mouse at actual points on the map. This function is similar to that described in Bahnij's Planner. Turnpoints and legs can be moved through the map interface by making the objects mouse sensitive.

Mouse sensitive lines are created by sending the mouse a message after it moves (:after :mouse-moves) to calculate its distance from the center of each line. If this distance is within a given range, the line is highlighted. When the mouse is clicked on a highlighted line, a new turnpoint is instantiated at the midpoint of the line, and this point can be moved by the same method described for moving mouse sensitive turnpoints.

Mouse sensitive turnpoints result when the :mouse-moves message finds the coordinates of the turnpoint within the highlighting range of the mouse. When the mouse is clicked on a highlighted turnpoint, the mouse grabs the turnpoint, erases the point display before the mouse moves, and redraws it after the mouse moves. The lines displaying mission legs between the turnpoints are erased and redrawn in the same manner. Another click of the mouse releases the turnpoint and its parameters are reset to the position at which it was released.

Mission parameters are computed using basic navigation algorithms. This data is placed in a planning database that interacts with the multiple tools in the planning system.

Planning Database. The planning database is stored in Flavors as instantiated objects or in KEE as frames. Both of these implementations use a frame structure and messages to access the values in the frames. KEE currently holds the unchanging mission and aircraft data, while Flavors stores the turnpoints, mission

legs, and threats. This dual structure was a consequence of rapid prototyping.

Originally implemented purely in KEE, the TMP required definition of new messages to perform proper manipulation of the objects. KEE uses active values for defining these new messages, and it seemed that these active values required unnecessary overhead. Flavors provided the same capabilities but at an easily accessed level of implementation. KEE will eventually be totally eliminated from the planner due to problems enumerated in Chapter VI.

Intelligence Analysis. Intelligence analysis is performed using a simple algorithm for determining the distance between lines and points. When any leg (a line) is within the effective range of a threat (a point), the planner notifies the user. Threats can be eliminated by removing them from the threat list. Elimination or relaxing threats is a method of the pilot telling the planner that he can defend against the threat, or that he does not wish to consider specific threat types in his analysis of the route. Knowledge is embedded in the threats of the best method of defeating the threat. This knowledge can be accessed when a novice planner finds that he cannot find a threat free route and needs to know how he can survive in the threat areas.

#### Summary

The detailed design of the mission planner presents a problem of manipulating many different functions in a method that allows different tools to access these functions. The interfaces to the functions are not complex; however, there are many inter-

faces between the tools. Not all the tools talk to each other (see Figure 4) and many of the functions for displaying various details of the mission involve modifying flavors to take differing interfaces into account. The success with which the interfaces were implemented will be discussed in the next chapter concerning system evaluation.

## V. System Evaluation

Evaluation of the planning system created for this thesis effort includes analysis of both the software and the planning capabilities of the TMP. This chapter will not include a detailed analysis of mission plans developed with the planner; such an analysis will be conducted when the TMP is employed in an operational squadron in 1987. This chapter will include an evaluation of the design methodology, a discussion of the human-computer interface principles affecting the design of the TMP, and a critique of the TMP's abilities.

### Design Methodology

An analysis of the design methodology shows both benefits and problems. The benefits of rapid prototyping are the ability to tailor the system to fit the needs of the pilots. In large software systems or subjects in which the programmer is not a domain expert, it is difficult to correctly specify the end product. If the software produced does not meet the needs of the users, a design revision must be implemented to meet new specifications. Revision may need to be performed a number of times before the end user is satisfied with the product. This is an example of how the traditional 'waterfall' design methodology becomes, in effect, prototyping. The difference in the two is that rapid prototyping develops the system in small increments where changes are much less extensive than the major revisions required for completed software products.

The problem with rapid prototyping in the TMP is the need to account for all the aspects of the end product as 'stubs' in the early prototypes. When these stubs are incrementally developed, the programmer finds that he has a large number of small software modules under development at one time, creating a problem of keeping track of the development of each module. If the programmer completes each module independently, he has a problem of interfacing these modules with each other, and design errors require greater revision in finished modules than in incomplete modules. The problem becomes one of bookkeeping in the prototype, and a tradeoff is made between the benefits of easy changes and the difficulty of building a program with many modules under development at once.

Well defined interface specification between the system modules will alleviate the majority of problems encountered with early design errors in rapid prototyping. In a project in which the system specifications are incomplete, the software engineer must provide for future changes by making stable interfaces between his modules. This stability means that the interface must be able to handle internal changes to the module without excessive changes to the interface itself. If the interface changes, then all modules that access that interface must change also, and there is a great deal of revision.

Interface stability in the TMP results from the functions provided in the implementation language (Lisp) and the Flavors package. Each part of the planner exists as some type of object (turnpoint objects, threat objects, window objects). The means of accessing these objects is standardized in the planning system.

through message passing. Each object receives some type of message that manipulates the object's attributes. By following this message passing standard for objects, the interface becomes stable and easy to manipulate by planning functions.

While it is easy to follow a specific design methodology such as message passing when putting together system modules, it is much harder to put together a system that interfaces well to the user. The difficulty in creating a suitable interface for a wide variety of users is a major problem in designing software systems.

#### Human-Computer Interface

The Human-Computer Interface is that aspect of the TMP that will insure its success or doom it to failure. In planning systems similar to the TMP, the interface to the pilots is under greatest scrutiny. Pilots will have no faith in a system that tells them which route to fly without giving a good reason for choosing that route. Pilots will be somewhat skeptical of systems that do not use some sort of graphical interface to the user, since pilots are more visually oriented than the average human. Considerations such as these must be reviewed when designing an interface between the human and the machine.

Woffinden has compiled a list of 12 items which deal with designing the human-computer interface (HCI) (Woffinden, 198x: 20). A prediction of the acceptance of the TMP by pilots can be made by studying how well the TMP adheres to Woffinden's design principles (Figure 5).

For the most part, the TMP falls neatly into compliance with the HCI principles. The purpose of the system is mission planning



for tactical (air-to-ground) missions. The end user is a fighter pilot who is going to drop bombs during his mission as opposed to shooting down other aircraft. The user is interested in spending as little time as necessary with the machine in order to discuss various aspects of the mission with his other flight members. The user prefers visual displays as opposed to text since the mind grasps information from a picture faster than it can analyze written words. These points were considered during the design of the TMP, and the resulting system is domain specific for air-to-ground missions, fast through its mouse interface, and presents information graphically.

1. Determine the purpose of the system
2. Know the user
3. Identify resources available
4. Consider human factors
5. Determine the interface language
6. Consider the environment of operation
7. Design for evolution
8. Optimize training
9. Accommodate levels of experience
10. Use selection vs. entry
11. Be consistent
12. Anticipate errors

Figure 5. Woffinden's HCI Design Principles

Human factors concerning system information and boredom are stressed. Since pilots want to know what the machine is doing at all times, there is a window displaying the current process (i.e., selecting a point, building a leg) and there is a constant graphical output when the machine is processing data (i.e., a point is displayed, a slot on the mission card is updated). The machine is totally controlled by moving the mouse and clicking

the mouse buttons, providing a major improvement over manual planning.

Resources available, the interface language, and the environment of operation should be considered together, since they all affect the type of machine upon which to implement the system. Resources available include fighter pilots, air-to-ground training manuals, and machine hardware. A machine supporting a language for rapid prototyping design, graphics, and a flexible interface language (in this case Lisp) is needed to adequately address the various aspects of mission planning. The choice of the tool used for implementation was discussed in Chapter IV. In considering the environment of operation, the Symbolics 3670 is not the machine of choice. The Symbolics' machine is superb for early prototyping, yet its physical size is too large for widespread use in squadrons. Chapter VI contains a detailed explanation of the effort for rehosting the TMP to a smaller computer.

The TMP was designed for evolution from its earliest beginnings. This is apparent in the name of the design methodology: rapid prototyping, also called evolutionary design. The consequences of rapid prototyping are annotated in the previous section.

Optimizing training, addressing various levels of experience, and using selection versus entry result from the menu system of the TMP. The system is designed for quickly developing a mission plan through the use of menus and the mouse. A trainee can immediately point with the mouse and select a menu, and can generate a mission plan rapidly by following the instructions displayed on the screen. It takes very little effort to become

an expert user of the system; the difference between an expert user and a novice is that experts are more precise with the mouse and know what information is in each menu. There are no commands that need to be input by the user; all commands are executed through selection with the mouse.

Consistency results from the use of menus as the only user input medium for the system, and through message passing. The user controls the entire program with the mouse, and as buttons are clicked or the mouse moves, messages are sent to the processor, which in turn calls messages created by the system designer. This scheme provides a means of standardization of both user inputs and function calls in the planner.

Anticipation of errors is the greatest failing of the TMP. In most systems designed through rapid prototyping, the end user performs analysis repeatedly and generally finds the majority of errors early in the system life-cycle. However, since the final evaluation of the TMP will not be conducted until 1987, some errors may lie undiscovered in the system. The author is not a domain expert in planning, and therefore, errors in performance cannot be anticipated. Other aspects of the TMP deserve analysis along with its error handling capabilities; therefore, a critique of the system is helpful to uncover both the merits and deficiencies of the planner.

#### TMP Critique

The TMP's map interface is a digitized terrain map of Nellis AFB, Nevada, and its training ranges. The TMP cannot be fully tested for accuracy since no pilot can currently plan a mission at AFIT in Ohio and then fly that mission at Nellis. This is due

to Nellis being the only available digitized terrain data in the early implementation of the TMP. The system will contain a library of maps for planning use when it is rehosted to its new environment in 1987.

There exists a problem with the implementation of mouse functions in the TMP. The TMP at times requires very deliberate clicks of the mouse when creating mission legs, or the display will not show the actual leg but a slightly skewed leg. This is due to the Symbolics processor 'getting behind' the mouse when many different functions are being processed. A redisplay of the map provides a correct display of the map and the mission, but this problem is annoying to system users and will be addressed upon rehosting the system.

Although the map and mouse show some deficiencies in the planner, the TMP as a whole is a useful development for tactical mission planning. Fast, precise turnpoint selection, modifying the route, and threat analysis are valuable to a pilot and difficult to perform manually. Revision of the route with the TMP takes on the order of 2 minutes, while revision of the route manually requires many more minutes. The TMP provides for contingency analysis through this revision, and the pilot has more than one opportunity to design the mission with the planner. These capabilities are important for future tactical mission planners, given the dense threat environments in which these missions must be flown.

## VI. Conclusions and Recommendations

### Conclusions

The mission planning system developed for this thesis shows the capability of designing and implementing a large software system through a rapid prototyping design methodology. By avoiding adherence to rigid specifications, the TMP quickly evolved to fulfill pilots' needs in mission planning. The design philosophy of building a working system, analyzing and including new requirements of the end users, and implementing the necessary changes provided a system that was essentially designed by pilots.

Two prototypes preceded the TMP: Major Robert Bahnij's thesis work (Bahnij's Planner) and the Intelligence Analysis Expert System (IAES). The merits of each of these prototypes were included in the TMP, and their deficiencies were addressed in the creation of the final prototype. Since the goal of rapid prototyping design is not operational software but requirements and specifications for such software, this thesis is the third step in designing an effective mission planning system.

The specific changes resulting from the prototyping methodology consist of numerous versions of the map interface, inclusion of a stores management system and a departure interface, and the ability to change the route from both the map and the mission card. The need for flexibility in the aircraft configuration and the mission route is not commonly addressed in other systems, and these areas result in new requirements for future mission planning systems. These requirements are subject to review and

revision in future prototypes as pilots use and critique the planner.

One consequence of prototyping is the ability of the knowledge engineer, who is not an expert in the domain for which he is developing software, to create a system that meets the needs of the end users. Bahnij claims that systems designed and produced by knowledge engineers who gather their information solely through interviews with domain experts "... are costly systems which fail to meet user needs" (Bahnij, 1985:VI-2). Such may be the case in software programs that, when created through rigid adherence to the traditional waterfall model, allow misinterpretation of the domain to cause errors that are left undetected until late in the software life-cycle, leading to catastrophic failure. However, prototyping gives the domain expert a more active role in system development by permitting the expert to analyze the system frequently, guiding the software designer through errors caused by the engineer's lack of knowledge of the domain.

This thesis is a prime example of the ability of a 'domain novice' to create a useable system solely through interviews and interaction with domain experts. The author has no experience with planning fighter missions other than research done for this thesis. By interviewing a domain expert (Capozzi, 1986) for initial specifications and then using another expert to frequently critique the resulting system (Lutz, 1986), a system that is effective and useful for tactical mission planning was created (Schoek, 1986).

## Recommendations

Operational Implementation. The driving factor behind this thesis was to take the hardware and software used for the TMP to an operational squadron in Europe. This project was conceived when Major General William Brechner, then commander of the 17th Air Force, USAFE, first evaluated the potential of Bahnij's prototype in late 1985 (Bahnij, 1985:VI-5). Since pilots in Europe were using antiquated equipment for mission planning and rapid prototyping showed promise of a quickly developed operational prototype, it was determined that research into mission planning systems and their use in operational squadrons should be conducted as a follow on to this thesis.

The current plan for the follow-on research is to have the TMP integrated into an operational squadron at Hahn AFB by mid 1987. The system will be tested and compared against hand-developed mission plans in order to check the TMP's accuracy and remove any bugs in the planner. After the TMP is verified and validated as a reliable system, it will be used for 12 months at the base for mission planning, and research will be conducted on the manner in which the pilots use the mission planner. The purpose of this research is to develop mission planning system requirements and specifications by analyzing the use of such systems in an operational environment. By noting the problems and benefits encountered as pilots use the TMP and other planning systems, a document will be written that expresses the true needs of operational squadrons. Such a document will encompass the whole of tactical mission planning, specifying the required interfaces to weather, intelligence, tasking, and other aspects of

mission planning. It is hoped that this document will assist in the creation of production planners for use throughout operational units.

Changing the Hardware Environment. After creation of three prototypes on the Symbolics 3670 under KEE, research needs to be conducted into a more economical environment for the operational TMP. It is apparent that the operational system should be independent of KEE, since KEE is a system development tool. There exists a considerable expense in both money (KEE costs approximately \$30,000) and computer performance (KEE is a large system written in InterLisp; there is a translation from InterLisp to ZetaLisp on the Symbolics machine that slows system operation) that can be saved by removing the TMP's reliance on KEE for storage of the knowledge base. It is proposed that the Flavors package be used to create a frame representation structure for storage of the knowledge base.

The Symbolics 3670 is a large, expensive (in the \$100,000 range) machine, and a smaller system capable of supporting the TMP will be more desirable in operational squadrons. Porting the TMP software to an adequate development environment for further prototyping on a computer that is easily integrated into current USAFE computer resources would prove beneficial to the overall project.

Map Research. The use of the mission planner is limited by the size and scale of the mission maps. A color map representing a 180 x 180 nautical mile area requires approximately 3,000,000 bytes of storage in two arrays which severely limits the amount of memory for other aspects of the TMP. Map displays representing



larger areas become prohibitively cumbersome unless a trade-off is made in map scale (and resolution). It is possible that the latest accomplishments in optical disk storage and the increase in on-line address space may solve these problems, and future research into this field is needed. The database and memory size problems may prevent widespread use of mission planners, since microcomputers are currently incapable of supporting these systems of this size, and other computers are either too large or too expensive at this time.

There exists a secondary map problem of digitizing aeronautical charts. The author found no easily accessible equipment for quickly storing aeronautical charts in computer memory for display on computer terminals. A surprising example of this lack of equipment exists in the Defense Mapping Agency (DMA). Instead of using computers to optically scan and digitize charts for the digitized feature analysis data (DFAD), DMA employs large numbers of people to pick objects off of maps and store them in the machines. While this human input allows storing of pristine information with no 'clutter' (i.e., contour lines, aeronautical flight corridors, and other non-terrain objects found on mission maps), it is extremely slow and laborious. Maps reproduced from DFAD data do contain useful cultural information; however, the clutter mentioned above is useful for mission planning and should be displayed on maps used in tactical mission planning systems. Therefore, the author recommends research into the areas of economically digitizing, compacting, and storing large aeronautical charts and the cultural information found on them.

Autonomous Planning. There has been a good deal of research into autonomous mission planners, and FLAPS and TEMPLAR are good examples of how such systems can be used operationally. It is suggested that research into autonomous planning be conducted by integrating Smith's Strategy Replanner into the TMP.

The proposed incorporation of the Strategy Replanner in the TMP requires two modifications to Smith's system: representation of the threat lethality contours (from a system such as FLAPS or through an added threat lethality processing capability) as threat vertices and analysis of terrain in the search process. The Strategy Replanner uses a set of four vertices to represent threats and does a search through these vertices for finding a route to the target. It should not be difficult to substitute the points used for the threat contours for the current vertices. If this is possible, the Strategy Replanner can simply choose the next higher probability of kill contour when it is required to relax threats. By using the preplanned points in Europe as vertices to be used in the search space, the Strategy Replanner can find a route to the FLOT with well defined turnpoints. As the replanner searches past the FLOT and through threats, it must be able to consider terrain in its remaining path to the target. The representation of terrain in the TMP/Strategy Replanner requires further research.

Contingency Analysis. The present TMP provides the pilot with a workstation in which he can change the state of the world to view different contingencies. The pilot can rearrange the known threat locations to explore the possibilities of different threats arrayed against him. As he does this the pilot makes his

own judgements of where to put the threats and how to deal with them. If the pilot is unexperienced in analyzing threats, these capabilities are not very useful. In such a situation, it would be desireable to have an expert system much like the IAES to assist the pilot in investigating different threat arrays. This system would have to be extremely fast in its analysis or it could not be used for operational analysis; however, an IAES would be beneficial as a training aide. The system would use the knowledge of experienced pilots as its knowledge base and be able to explore many scenearios. Research into contingency analysis systems should be conducted to explore their use in training and operational planning systems.

Briefing The TMP. The ability to save mission plans exists in the TMP. An extension of this would be a system that allows the pilot to 'brief' the TMP with his requirements for each mission. The pilot would be programming the planner to calculate missions tailored specifically to his desires. Pilots would brief the system by saving their preferences in a knowledge base that contains the method in which pilots programmed previous missions. When the pilot has a planning situation he has not encountered before, his instructions to the computer are saved in this knowledge base. For example, Pilot A desires to traverse a high threat area for a short time rather than a low threat area for a long time. Pilot B prefers to fly through the low threat rather than the high threat. In future situations in which the route must traverse terrain under threat, the computer would suggest that Pilot A fly the shortest route that traverses the threat. The computer would suggest a route to Pilot B through

the least potent threat even though the route would spend a longer time under threat. Information in this planning history knowledge base can be used as heuristics for autonomous planning algorithms. Continued research into these ideas for expert system mission planning aids is recommended, and analysis of the resulting prototype systems will further the state of the art in planning tools.

#### Summary

The flexibility of the planning environment of the TMP permits the exploration of many different flight plans and an overall increase in pilot situation awareness. Research into the areas of computerized mission planning systems holds the promise of providing pilots with the time they need to brief, create, and evaluate tactical mission plans in an environment in which pilots' response time to tasking orders is shrinking. This thesis, its continuing research and development, and the project's operational implementation are intended to make a useful contribution to both pilots in planning and executing tactical missions and the research community in providing an operational assessment of mission planning systems.

## BIBLIOGRAPHY

- Bahnij, Maj Robert B. A Fighter Pilot's Intelligent Aide For Tactical Mission Planning. MS thesis GCS/ENG/85D-1. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, December 1985.
- Barr, Avron and Edward A. Feigenbaum. The Handbook of Artificial Intelligence, Volume 1. Los Altos, California: William Kaufman, Inc., 1981a.
- The Handbook of Artificial Intelligence, Volume 2. Los Altos, California: William Kaufman, Inc., 1981b.
- Bradshaw, 2Lt Jeffrey S. "Threat Analysis Design Report" Unpublished report for the Intelligence Analysis Expert System. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, 27 April 1986.
- Capozzi, Maj Rocky, F-16 Pilot, 496th Tactical Fighter Squadron. Personal interviews. Hahn AB, West Germany, 22 through 30 July 1986.
- Cohen, Paul R. and Edward A. Feigenbaum. The Handbook of Artificial Intelligence, Volume 3. Los Altos, CA: William Kaufman, Inc., 1982.
- Engelman, Carl and others. "KNOBS: An Integrated AI Interactive Planning Architecture," AIAA Computers In Aerospace IV Conference. 450-462. AIAA Press, New York, 1983.
- "Fairchild Developing MAPS System To Speed Low-Level Mission Planning," Aviation Week and Space Technology: 97 (6 October 1986).
- Fikes, Richard and Tom Kehler. "The Role of Frame-Based Representation in Reasoning," Communications of the ACM, 28: 904-920 (September 1985).
- Foley, James D. and Andries Van Dam. Fundamentals of Interactive Computer Graphics. Reading, MA: Addison-Wesley Publishing Company, 1984.
- Foley, Capt Jeffrey W. "Problem Assessment Report: Tactical Air-to-Ground Threat Analysis Knowledge Based System." Unpublished report for the Intelligence Analysis Expert System. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, 27 April 1986.
- Foxwell, Col Mark B., Assistant Deputy Commander for Operations. Correspondence with HQ USAFE/DOOS concerning software for F-16C DTCL/R. Hahn AB, West Germany, May 1986.

- Goldberg, Adele. Smalltalk-80 The Interactive Programming Environment. Reading, MA: Addison-Wesley Publishing Company, 1984.
- Hearn, Donald and M. Pauline Baker. Computer Graphics. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1986.
- Ingalls, Daniel H. "The Smalltalk-76 Programming System, Design, and Implementation," Conference Record of the Fifth Annual ACM Symposium on Principles of Programming Languages. 9-17. January 1978.
- "Lisp Machine Summary 3600 Edition," Symbolics, Inc., August 1983.
- Lutz, Rollin J., A-10 Pilot. Personal Interviews. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, March through December 1986.
- Pascoe, Geoffrey A. "Elements of Object-Oriented Programming," Byte, 11: 139-144 (August 1986).
- Phillips, Capt Geoffrey S. "Knowledge Acquisition Report for a Tactical Air-to-Ground Threat Analysis System." Unpublished report for the Intelligence Analysis Expert System. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, 5 May 1986.
- Rainbolt, Stephen M. Force Application Management With AUTOPATH. Final report, Contract No. F61546-84-C-0088. Systems Control Technology, Inc., Palo Alto, CA, 30 October 1985.
- Rich, Elaine. Artificial Intelligence. New York: McGraw-Hill Book Company, 1983.
- Rockmore, Joseph and others. "Route Planning Aid: System Specification." Report 6459-400-03 prepared for PAR Technology Corp. Systems Control Technology, Inc., Palo Alto, CA, September 1983.
- Rupp, Capt Alex, F-16 Pilot, 313th Tactical Fighter Squadron. "Min Time Tasking Program (Version 18)." Computer program. Hahn AFB, West Germany, 1986.
- Shapiro, Daniel G. and others. TEMPLAR (Tactical Expert Mission Planner) Design. Report No. RADC-TR-84-134. Rome Air Development Center, Griffiss AFB, NY, June 1984.
- Schoeck, James, A-10 Pilot. Personal Interview. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, 22 November 1986.
- Simpson, Capt Leslie. Project Officer, Mission Support System program. Telephone interview. Tactical Air Forces Interoperability Group, Langley, VA, 6 Oct 1986.

Smith, Wesley M. An Artificial Intelligence Aid for B-1B Mission Replanning. MS thesis GCE/ENG/86D-7. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, December 1986.

Wilensky, Robert. LISPcraft. New York: W. W. Norton and Company, 1984.

Winston, Patrick Henry. Artificial Intelligence (Second Edition). Reading, MA: Addison-Wesley Publishing Company, 1984.

Winston, Patrick Henry and Berthold Klaus Paul Horn. Lisp (Second Edition). Reading, MA: Addison-Wesley Publishing Company, 1984.

Woffinden, Duard S. Interactive Environment for a Computer-Aided Design System. Master's Thesis, Naval Postgraduate School, Monterey, CA, June 1984.

Additional Sources Not Referenced in the Thesis

- Bowyer, Adrian, and John Woodwark. A Programmer's Geometry. London: Butterworths, 1983.
- Charniak, Eugene and others. Artificial Intelligence Programming. Hilldale, New Jersey: Lawrence Erlbaum Associates, Inc., 1980.
- Gilmore, John et al. "Knowledge Based Tactical Terrain Analysis," SPIE Digital Image Processing Conference, San Diego, CA (August 18-23, 1985).
- Gilmore, John F. and Antonio C. Semeco. "Terrain Navigation Through Knowledge-Based Route Planning," International Joint Conference on Artificial Intelligence, Los Angeles, CA (August 1985).
- . "Autonomous Route Planning Through Non-Uniform Terrain," IEEE Computer Vision and Pattern Recognition, San Francisco, CA (1985).
- Grossberg, L. "Terrain Masking Algorithm," Technical Memo, Systems Control Technology Inc, Palo Alto, CA (November 1981).
- Harmon, Paul and David King. Expert Systems: Artificial Intelligence in Business. New York: John Wiley and Sons, Inc., 1985.
- Jardine, T. J. and S. T. Shebs. "Knowledge Representation in Automated Mission Planning," Proceedings of the International Joint Conference on Artificial Intelligence. (1980).
- Mostow, Jack. "Forward: What is AI? And What Does It Have to Do with Software Engineering?," IEEE Transactions on Software Engineering, SE-11: 1253-1256 (November 1985).
- Nilsson, Nils J. Principles of Artificial Intelligence. Palo Alto, California: Tioga Publishing Company, 1980.
- Sacerdotti, Earl D. "Planning in a Hierarchy of Abstraction Spaces," Artificial Intelligence, 5: 115 (Summer 1974).
- . "Problem Solving Tactics". AI Magazine. 7-15, (Winter 1980-81).
- Stefik, Mark. "An Examination of a Frame-Structured Representation System," Proceedings of the Sixth International Joint Conference on Artificial Intelligence. 845-852. Tokyo, Japan, 1979.



- . "Planning and Meta-Planning," Artificial Intelligence,  
16: 141-169 (May 1981)b.
- . Planning with Constraints. PhD dissertation. Computer  
Science Department, Stanford University, Stanford CA, January  
1980 (STAN-CS-80-784).
- Sussman, G.J. and G. L. Steele Jr. "CONSTRAINTS - A Language for  
Expressing Almost-Hierarchical Descriptions," Artificial  
Intelligence, 14: 1-39 (January 1980).
- Waldinger, R., "Achieving Several Goals Simultaneously," SRI  
Artificial Intelligence Center Technical Note 107 (July  
1975).
- Waterman, Donald A. A Guide to Expert Systems. Reading, MA:  
Addison-Wesley Publishing Company, 1986.
- Wilkins, D. E. "Domain-independent Planning: Representation and  
Plan Generation," Artificial Intelligence, 22: 269-301  
(April 1984).
- Winograd, Terry. "Extended Inference Modes in Reasoning by Com-  
puter Systems," Artificial Intelligence, 13: 5-26 (1980).

## Appendix A: An Early Version of the TMP

This appendix contains figures of an early version of the TMP. The system has gone through repeated upgrades since the photographs in this appendix were taken. This appendix will describe the TMP and its displays.

Figure 6 shows the flight card display. Two turnpoints have been selected, and the information about the points and legs between the points has been displayed on the card. Figure 7 shows a completed mission, with all leg parameters filled in. The current version of the TMP has a flight card that is similar to these figures except for 4 slots for the legs and turnpoints.

Figure 8 shows the departure interface. Since the TMP does not contain the map of Hahn AB, the departure interface is not linked into the system. Upon upgrading the TMP for use at Hahn, this interface will be used for selecting the first few turnpoints.

Figure 9 contains the Stores Management System panel. This window allows the user to select his configuration from the menu displayed in the middle of the figure. The selected configuration is "Bombs" with Snakeye bombs loaded on the aircraft. If the user wishes to reconfigure the aircraft, he can choose a different SCL from the menu, or he can selectively reconfigure each panel. Notice the single Snakeye on Station 3 in Figure 9. The user changes this single bomb to 2 Mark 82 bombs by selecting the Stores Menu (Figure 10), choosing the Bombs Menu (Figure 11), and selecting a Slant-2 rack from the Rack Menu (Figure 12). The result is 2 Mark 82 bombs loaded on Station 3 (Figure 13).

The color monitor (Figure 14) displays the map and other panels. The notification window, located in the upper left hand corner of the color screen, is shown in Figure 15. The AGL/Speed panel (Figure 16) allows the user to set his distance from the ground (AGL) and his airspeed as he plans his mission. Below the AGL/Speed panel is the Color SMS Window, which is a color display of the current configuration (Figure 17). When the black and white SMS panel changes, the color SMS window changes also, giving a pilot a constant reminder of his weapons loading. Below the SMS panel is the map legend, giving simple elevation information for the terrain map.

Figure 18 shows a typical threat array. A threat display consists of a bitmap displaying the type of threat (SAM or AAA) and the threat name (ZSU, SAM 6). The circles represent the maximum effective range of each threat. The FLOT is displayed in Figure 18 by the thick curving line through the figure.

Figure 19 shows the Planning Choices menu. The user selects his planning functions from this menu, and selects turnpoints on the terrain map surrounding the menu. The map is similar to a relief map. Figure 20 shows both the threat menu and part of a mission route. The threat menu currently contains more commands than are displayed in this figure (i.e., Analyze, Degrade, and Move threats). The turnpoint in Figure 20 is selected by simply clicking the mouse at a point on the screen when executing the function "Build Mission Legs" from the Planning Choices menu of Figure 19. Figure 20 shows the Display Control menu and the starting point (the circle in the lower right center of the figure).

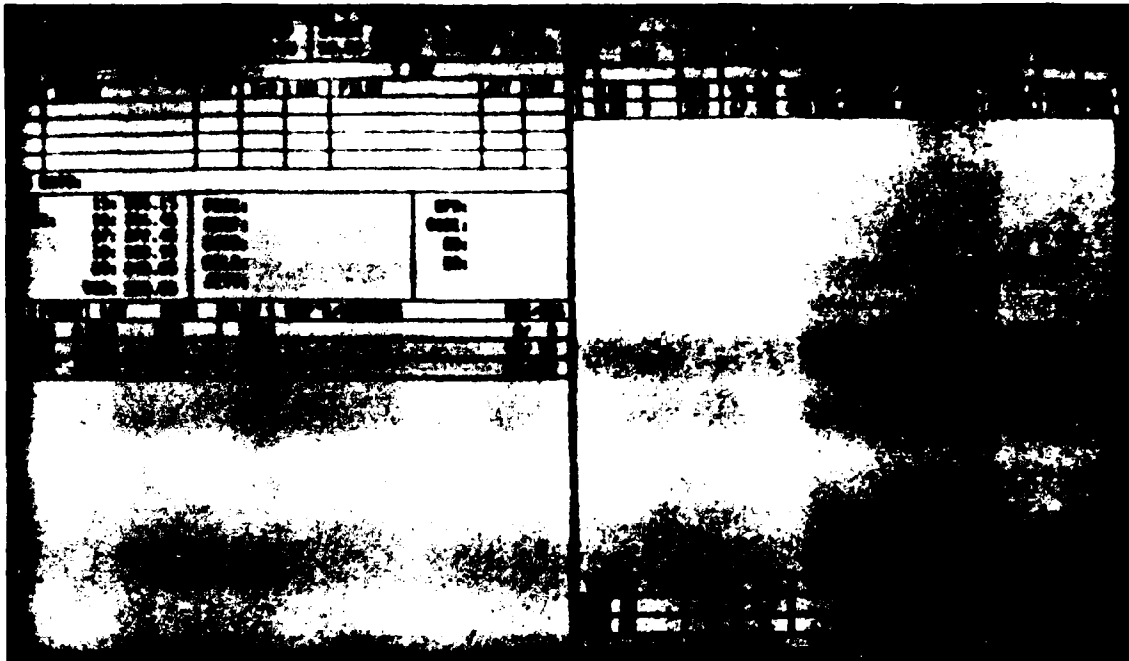


Figure 6. Incomplete Mission

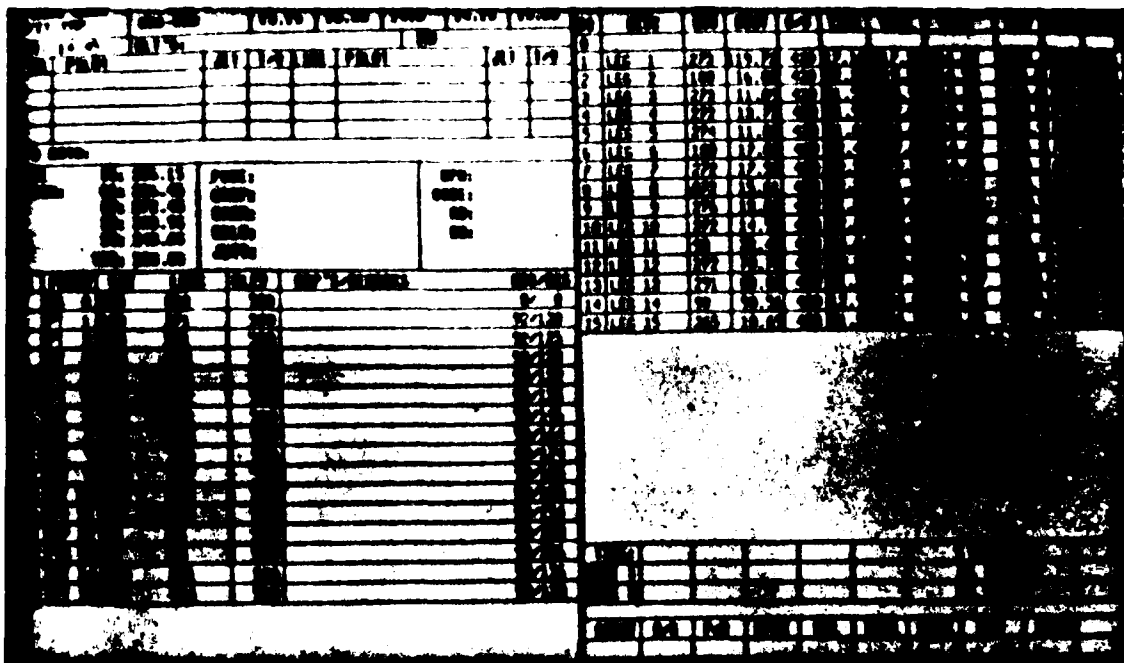


Figure 7. Completed Mission

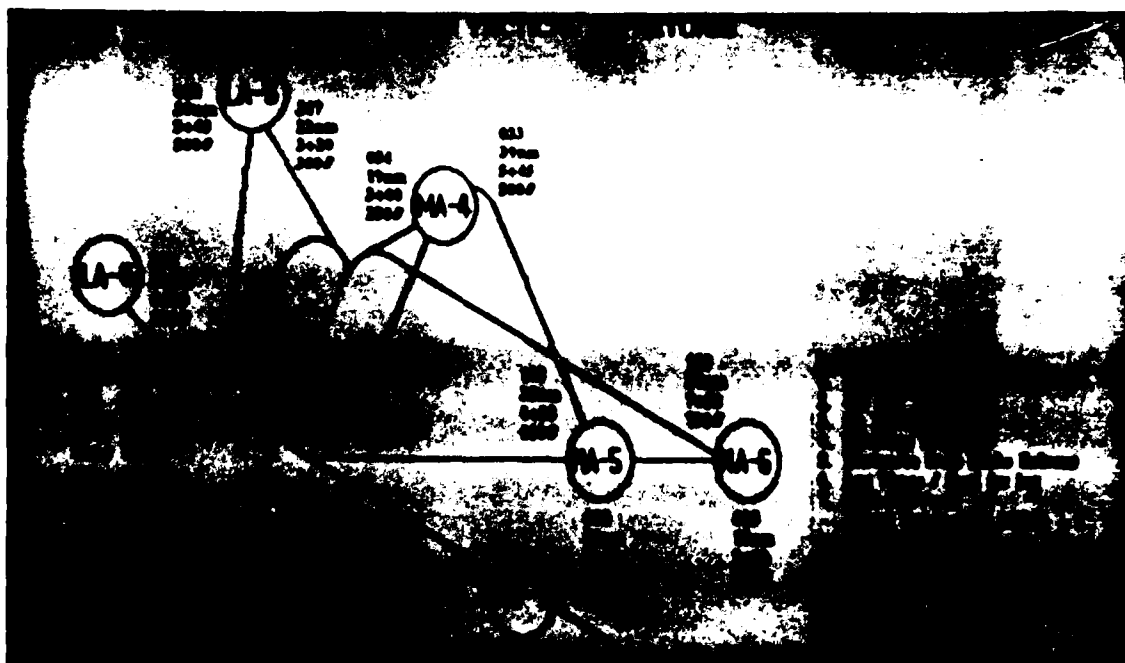


Figure 8. Departure Window

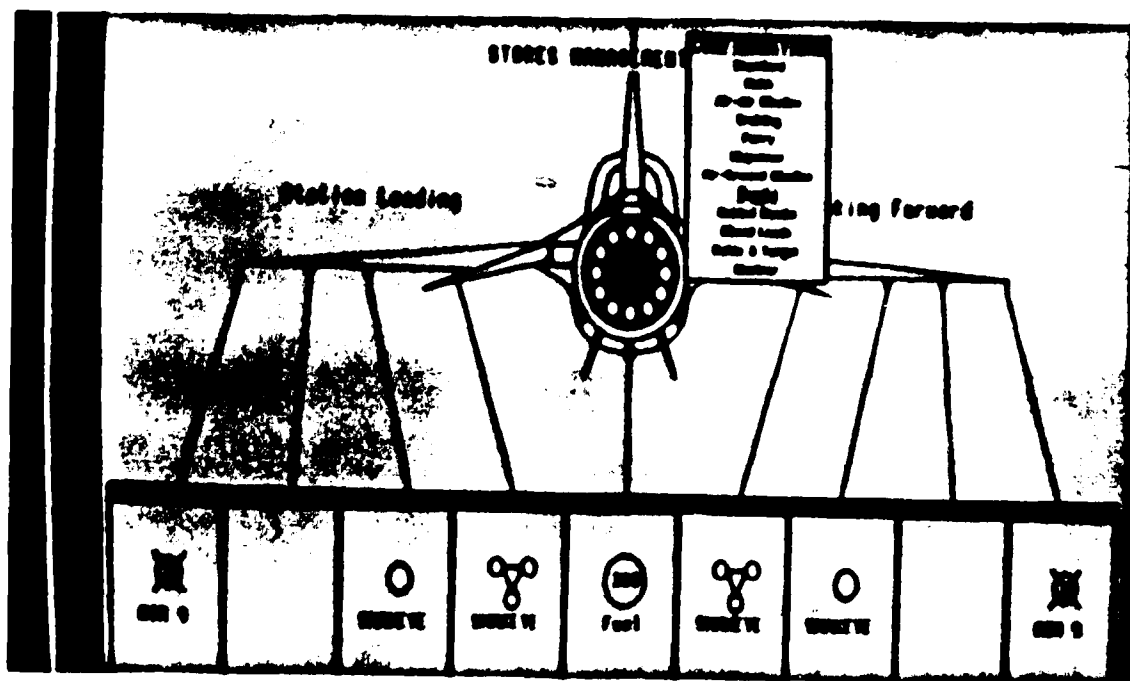


Figure 9. Stores Management System



Figure 10. Stores Menu



Figure 11. Bomb Menu

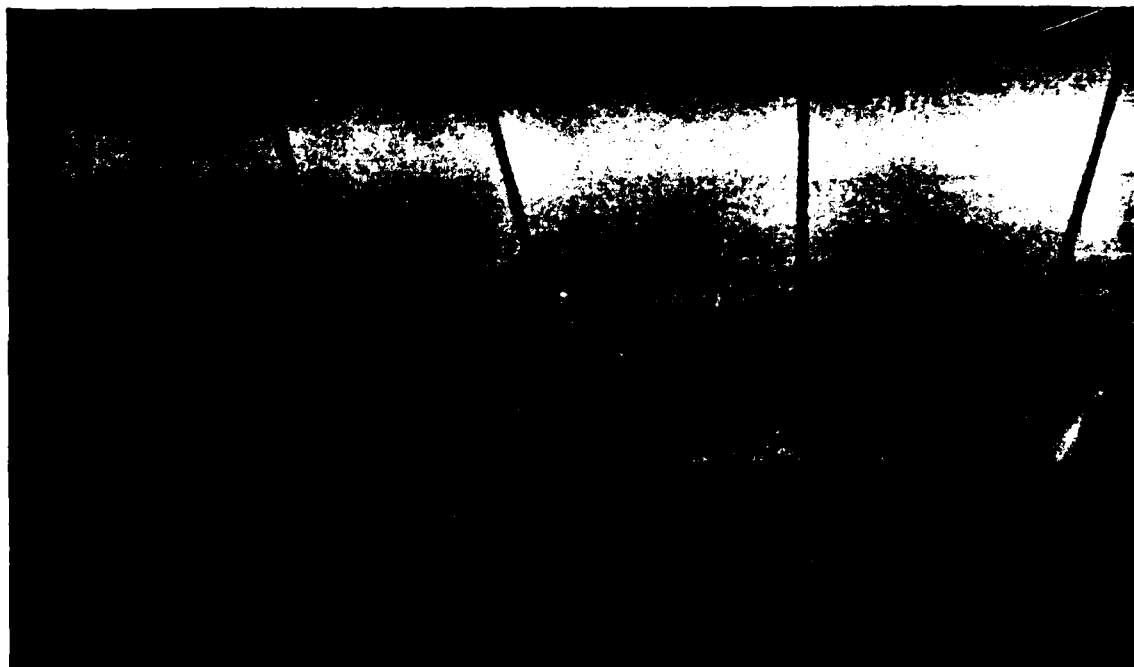


Figure 12. Rack Menu

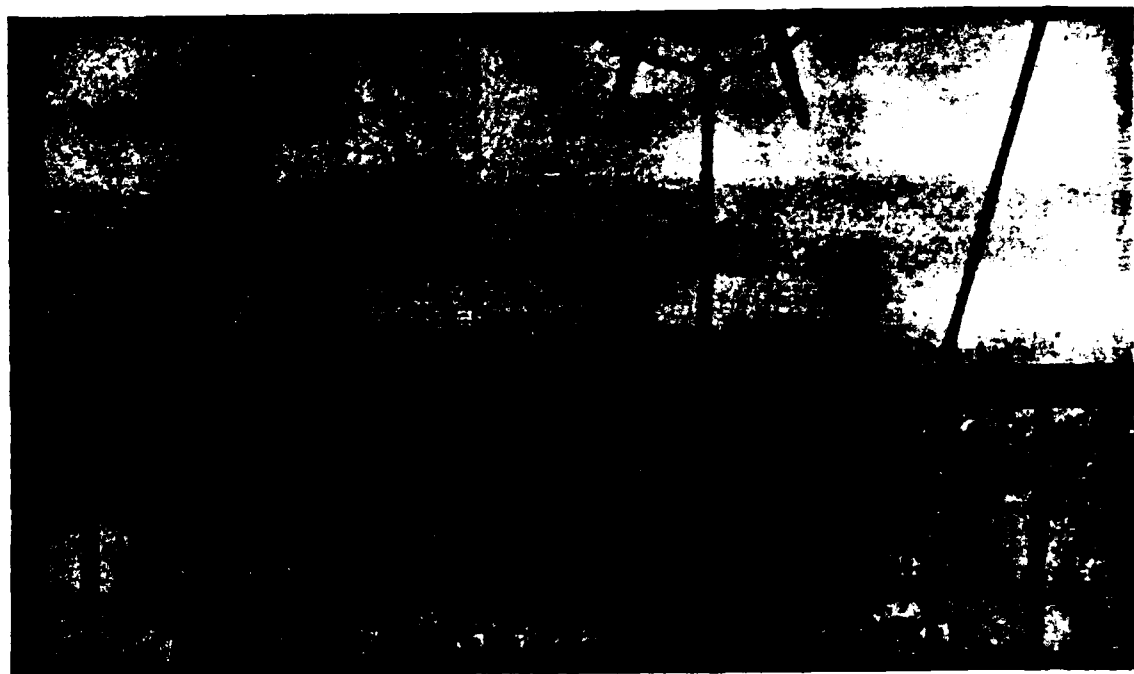


Figure 13. New Station Loading



Figure 14. Color Display

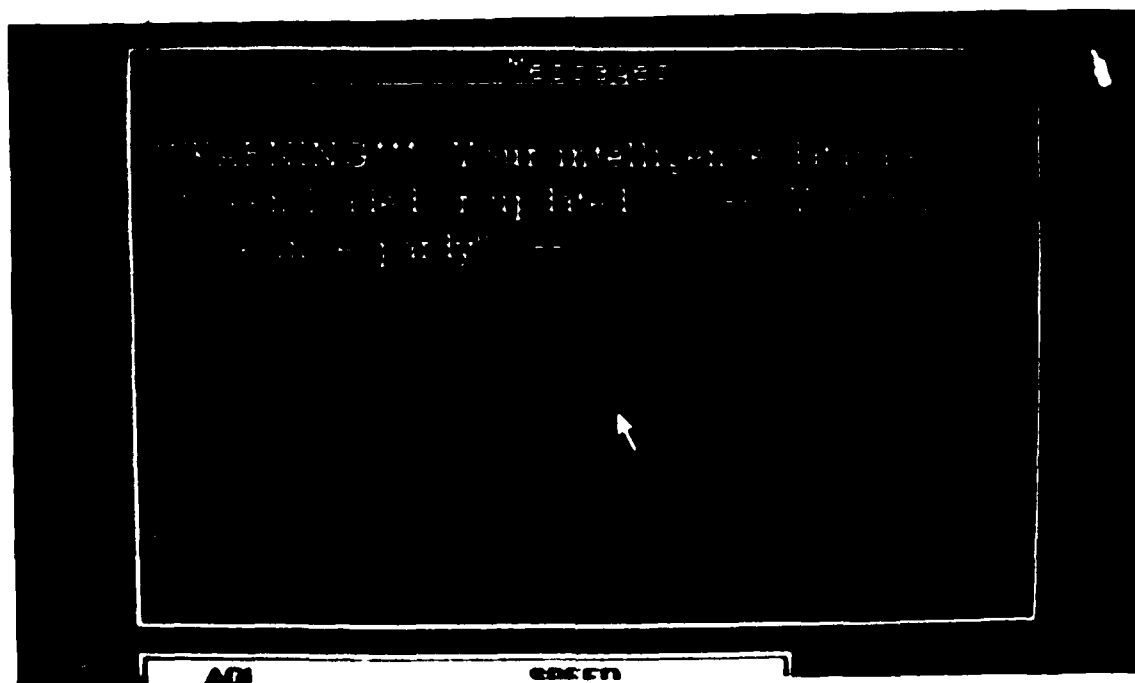


Figure 15. Message Window



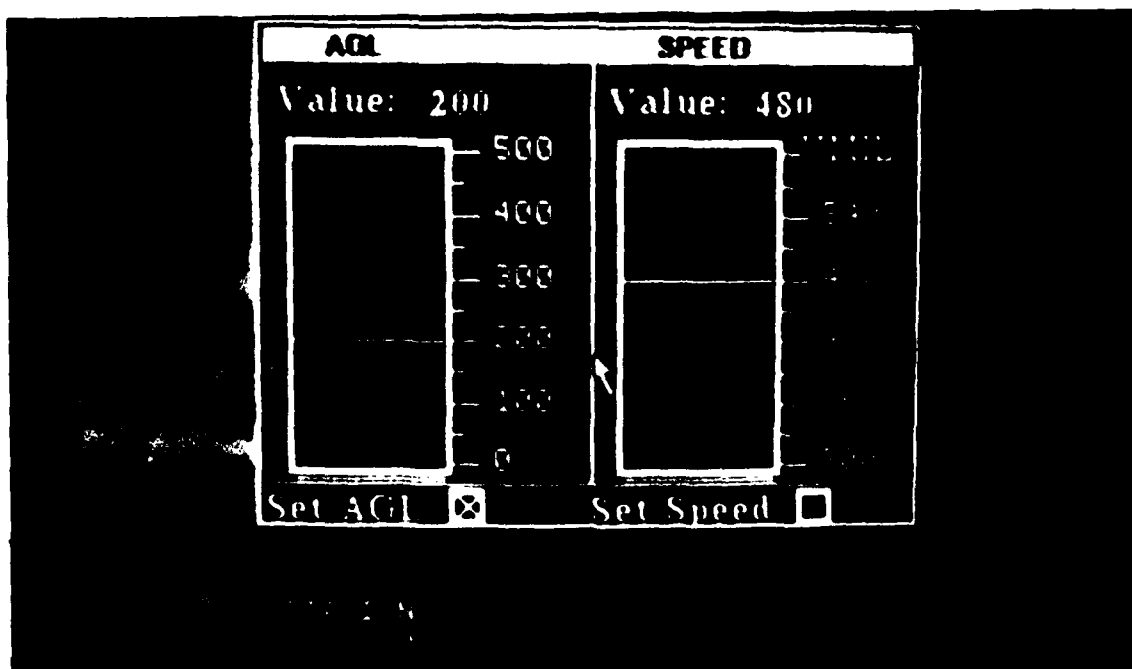


Figure 16. AGL/Speed Panel

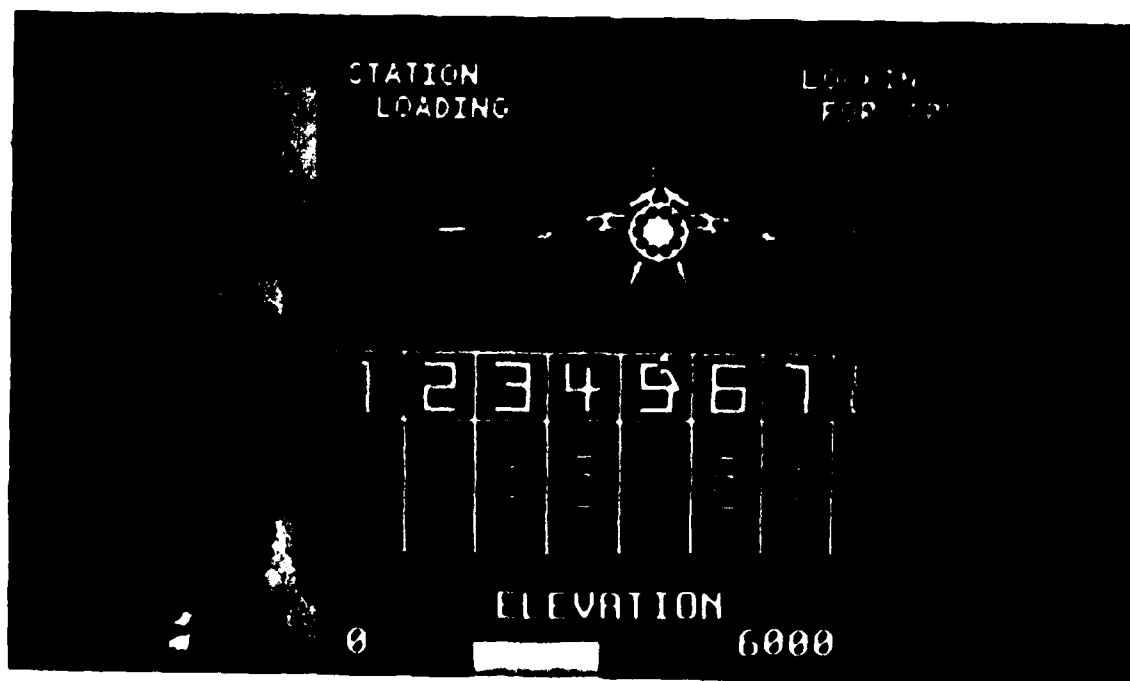


Figure 17. Color SMS Panel



Figure 18. Typical Threat Array

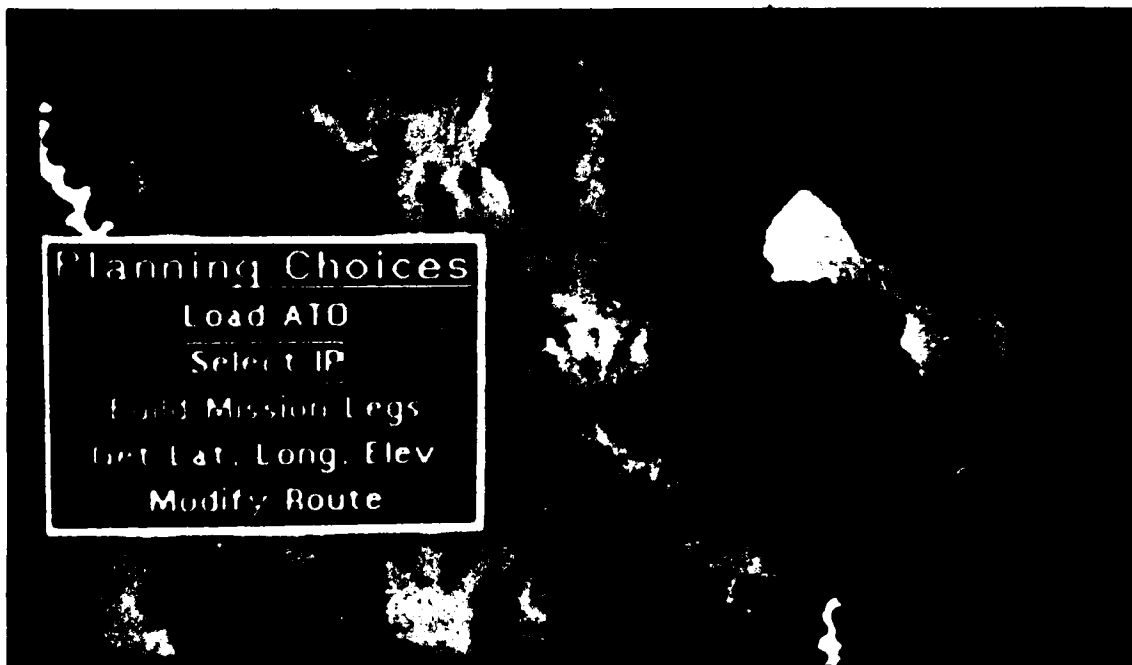


Figure 19. Planning Menu

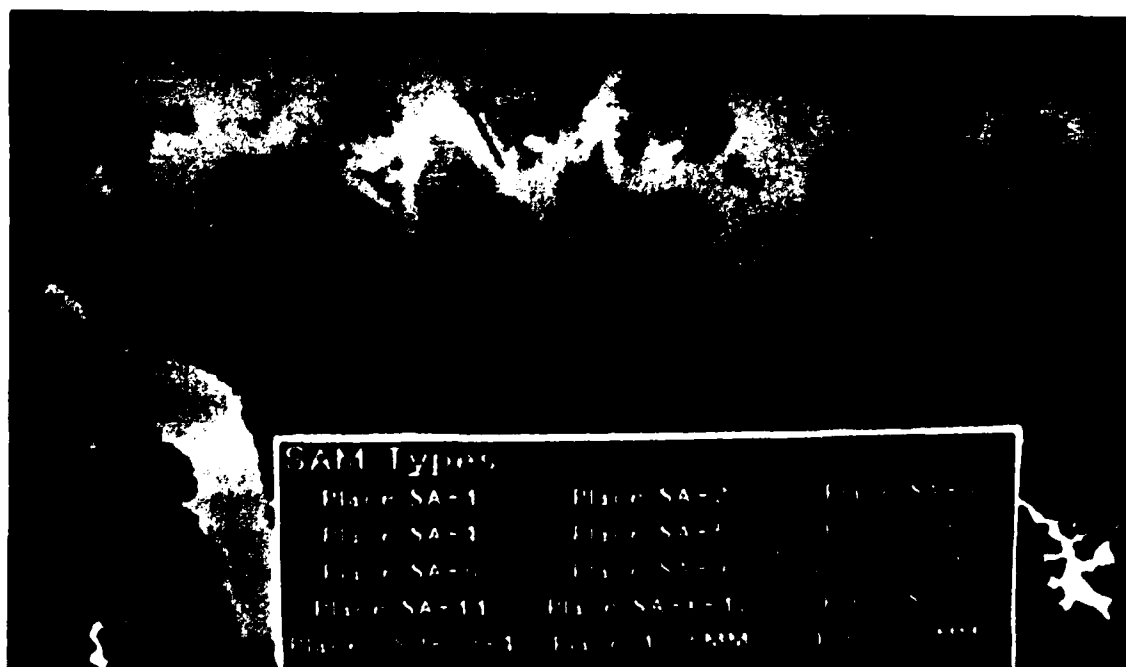


Figure 20. Threat Menu



Figure 21. Display Menu

## Appendix B: TMP User's Manual

The Tactical Mission Planning system developed for this thesis is very easy to use; the only requirement of the user is precision when using the mouse. This appendix should provide the necessary training for any individual to use the system. This appendix assumes that the hardware running the system is not booted (not up and running) but the machine is turned on. All commands that are typed on the keyboard are in double quotes, with a <cr> symbol meaning carriage return. A mouse click refers to pressing and releasing the specified mouse button. 'Run bars' are small horizontal lines centered at the bottom of the black and white screen; when these lines appear and flash on and off, the system is processing commands. It is usually best (but not necessary) to wait until the run bars stop flashing before entering the next command. Also, when clicking the mouse, the user should deliberately press the button while not moving the mouse. Moving the mouse causes the mouse process to 'smear' the mouse click, and precision is lost.

Some standard points to remember are as follows: the large letters under the map window explain the current process and should be read and followed at all times; a right mouse click usually aborts the current process; "Change Screen" moves the mouse back and forth between monitors; and the user should be slow and deliberate when using the system until he feels comfortable with the planner.

The TMP resides on a Symbolics 3600 machine at AFIT. It is necessary for this machine to be running KEE; therefore, the user should type "b >kee.boot<cr>". The system loads in the KEE software, and after a few minutes there will appear 3 square windows on the screen. The cursor will blink in the window titled "Lisp Listener". The user now must enter "(login 'jsb :host 'spl)" to load in the TMP software. The system will generate windows on both the color and black and white (b&w) screens, and after a few minutes a logo will appear. The system is now loaded and ready to plan missions.

The first thing that must occur for planning is receipt of the ATO. The user will see the mouse on the color screen. Clicking on the square next to the word "Plan" will produce the planning menu, and the user loads the ATO by choosing the "Load ATO" option from that menu. The system will display the start point (a circle in the lower right corner of the map), the target (a triangle in the upper left corner of the map), and LLTR points (circles between the start and target) contained in the ATO on the map.

After loading the ATO, the TMP will notify the user that all planning functions will be disabled until the threat database is loaded. To load this data, click on the "Display" square, and choose "Display Current Intelligence" with the mouse. The color screen will now show the FLOT and current threat array, and planning can continue. The threat array needs to be loaded only once, and after the system reads the intelligence data it will not prevent planning for future missions.

The user will notice the Stores Management System window on the b&w monitor. The TMP has loaded the aircraft with a configuration given in the ATO. If the user desires, a new configuration can be loaded by clicking on "Change Screen" on the color window to move the mouse to the b&w window. Now choosing "Standard SCL" on the SMS window will produce a menu of typical configurations. Selecting any of these configurations will load that SCL into the system.

The user can change specific stores on the aircraft by clicking the mouse in the center of any of the windows labeled "Station N" where N is a number from 1 to 9. A menu will appear in the window, and the user can select the new station loading. Some menus (such as the "Bombs" menu) produce new menus (see Appendix A). The user can either make choices in these nested windows to change the current loading, or he can move the mouse out of the menu, in which case the menu disappears and the station remains unchanged.

Choosing the box labeled "Change Screen" on the SMS window places the mouse back on the color screen, and planning can continue. Selecting the "Plan" menu and choosing "Select II" allows the user to choose the initial point. The system expects the user to click the mouse at some location close to the target (within one inch of the target triangle). A square will appear which represents the II.

Whenever the user wants to change the speed or altitude of the aircraft he simply clicks the "Change Speed" or "Change Alt" box on the AGL/Speed panel. After clicking on the box, the user should move the mouse to his desired speed/Alt and click the

mouse again. The system will not allow going faster than VMil speed or lower than 50 feet AGL. An attempt to do so will enter a default value for the current speed/AGL.

The user at this point should select "Plan" and choose "Build Mission Legs" from the planning menu. The mouse will now trail a line from the current mouse location to the last selected point, which is currently the start point. The user can choose turnpoints by clicking the left mouse button at any point, or he can enter an LLTR by clicking the mouse within one of the 3 closest circles to the start. If the user chooses an LLTR, he should wait (about .5 seconds) until the mission card on the b&w screen displays all the LLTR points and legs before continuing. It is usually best to choose an LLTR, since most missions are controlled by the ACO on the friendly side of the FLOT.

When the user wishes to select the IP as the next point, he simply clicks the middle mouse button 2 times in rapid succession (a double click). The system will make a leg from the last selected point to the IP, then to the target. This double click must occur at some point in the mission else the system will consider the mission to be incomplete, and an error will occur.

The user can now choose points back across the FLOT to the friendly side and home. If an LLTR endpoint is selected, the LLTR will become a part of the mission in reverse order. To complete the mission, a double left click takes the user from whatever point he last chose to the start point and updates the flight card.

If the mission consumes too much fuel, the flight card will flash and a "Busted Bingo Constraint" notice will appear on the

screens. The user must now modify his route to bring the fuel consumption within limits. This is done by choosing "Plan" and selecting "Modify Route" from the menu. During modification, the turnpoints will highlight when the mouse gets within 20 pixels of them, and clicking the mouse (left or middle button) on a highlighted turnpoint grabs the turnpoint with the mouse. Another mouse click releases the turnpoint at the current mouse location, and the mission card updates. To quit modifying points, click the right mouse button.

The threat array can be analyzed by choosing the "Threat" square on the map window and selecting "Analyze Threats" from the threat menu. The system will find all legs that are within the lethal radius of a threat and notify the user which legs are under threat. The threat menu also allows the user to place threats, move threats, or degrade threats.

By selecting "Place xxxx" from the threat menu, where xxxx is a threat name, the user can add new threats to the threat array. The user will place whatever threat he has chosen wherever he clicks the mouse. The user can continue to place threats until a right click ends the threat placement process.

The user can move threats by selecting "Move Threat" from the threat menu. When the mouse is close to a threat, a reverse video circle will appear over that threat. By clicking on this circle, the mouse grabs the threat (similar to moving a turnpoint). Moving the mouse to a new location and clicking a button releases the threat, and another threat can be moved. A right click aborts the threat movement process.



### Vita

Jeffrey S. Bradshaw graduated with honors from East Texas State University at Commerce, Texas, in December, 1984. He was commissioned into the United States Air Force in May, 1985. His first assignment was to the Air Force Institute of Technology, Wright-Patterson AFB, Ohio. After graduation from AFIT in December, 1986, Lieutenant Bradshaw will be assigned to Headquarters, 17th Air Force (USAFE) at Sembach AB, West Germany as a computer systems engineer.

Lieutenant Bradshaw is married to the former Melissa Michelle Vaught of Kansas City, Missouri. The Bradshaws have a daughter, Jessica Michelle, age 12 months.

A178871

DD Form 1473, JUN 86

SECURITY CLASSIFICATION OF THIS PAGE

Unclassified

Unclassified

Abstract

✓  
A prototype tactical mission planning system is described which provides pilots with automated tools for developing air-to-ground mission plans. This thesis is a continuation of the Fighter Pilot's Intelligent Aide For Tactical Mission Planning.

The Tactical Mission Planner (TMP) contains interfaces to planning and intelligence databases and performs rudimentary threat analysis. Extensive use of graphical displays provide the user with an interactive visual environment for generating mission routes from a starting base to the target and back to the home base. The TMP employs a rapid prototyping design methodology coupled with object oriented programming. This design strategy permits the knowledge engineer to produce an effective planning system without expert knowledge of the planning domain.

END

5-87

DTIC